UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO DE CIÊNCIAS AGRÁRIAS
DEPARTAMENTO DE COMPUTAÇÃO

JAKSON JOSÉ SCHEIDER LAQUINI

MODELAGEM E IMPLEMENTAÇÃO DA META-HEURÍSTICA CLUSTERING SEARCH PARA RESOLUÇÃO DO PROBLEMA DE ESCALONAMENTO DE TRIPULAÇÕES

ALEGRE - ES
2014

JAKSON JOSÉ SCHEIDER LAQUINI

MODELAGEM E IMPLEMENTAÇÃO DA META-HEURÍSTICA CLUSTERING SEARCH PARA RESOLUÇÃO DO PROBLEMA DE ESCALONAMENTO DE TRIPULAÇÕES

Trabalho de conclusão de curso apresentado ao Departamento de Computação do Centro de Ciências Agrárias da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.
Orientador: Prof. Dr. Geraldo Regis Mauri.

ALEGRE - ES
2014

JAKSON JOSÉ SCHEIDER LAQUINI

MODELAGEM E IMPLEMENTAÇÃO DA META-HEURÍSTICA CLUSTERING SEARCH PARA RESOLUÇÃO DO PROBLEMA DE ESCALONAMENTO DE TRIPULAÇÕES

Trabalho de conclusão de curso apresentado ao Departamento de Computação do Centro de Ciências Agrárias da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Aprovado em 09 de Dezembro de 2014.

COMISSÃO EXAMINADORA

Prof. Dr. Geraldo Regis Mauri
Universidade Federal do Espírito Santo
Orientador

Prof. M.Sc. Antonio Almeida de Barros Junior
Universidade Federal do Espirito Santo

Prof. M.Sc. Edmar Hell Kampke
Universidade Federal do Espírito Santo

A meus pais, José e Tereza que me apoiaram e me deram força durante toda graduação.

AGRADECIMENTOS

A Deus, por guiar e permitir que chegasse até aqui.

A meus pais José e Tereza, que mesmo com tantas dificuldades, sempre me apoiaram e me deram forças para nunca desistir.

A minha irmã Betânia, que sempre esteve ao meu lado, incentivando e aconselhando nos momentos mais difíceis.

Aos familiares, que sempre torceram por mim.

Aos professores do DCOMP, pelos ensinamentos e experiências transmitidos.

Ao professor Geraldo, pela orientação, dedicação, correções e incentivos fornecidos durante todo desenvolvimento do trabalho.

A Ana Carla, por todo companheirismo, carinho e paciência durante todos esses anos.

A todos os amigos construídos durante este período de faculdade, por estarem sempre juntos nas noites viradas, trabalhos, festas e por todos os momentos vividos, nunca serão esquecidos.

A todos que torceram por mim, muito obrigado.

RESUMO

O Problema de Escalonamento de Tripulações (PET) consiste em encontrar uma melhor distribuição de tarefas às tripulações considerando algumas restrições trabalhistas e empresariais. O objetivo é encontrar uma distribuição que garanta que todas as viagens realizadas por uma empresa sejam executadas com o menor custo possível, em que para alcançar esse baixo custo deve-se eliminar o tempo sobreposto e reduzir a hora extra, o tempo ocioso e o número de tripulações necessário para executar todas as tarefas da empresa. Este trabalho apresenta uma meta-heurística híbrida conhecida como *Clustering Search* (CS) para resolução do problema. Para medir o desempenho do método proposto, foram utilizados problemas encontrados na literatura, e os resultados obtidos foram comparados a outros métodos já estudados para a resolução do PET.

Palavras-Chaves: *Clustering Search;* Problema de Escalonamento de Tripulações; Meta-heurísticas.

MODELING AND IMPLEMENTING A CLUSTERING SEARCH METAHEURISTIC FOR SOLVE A CREW SCHEDULING PROBLEM

ABSTRACT

The Crew Scheduling Problem (CSP) consists of finding a better distribution of tasks to crews considering some labor and business restrictions. The goal is to find a distribution that ensures that all trips assigned to a company can be performed with the lowest cost. The low cost should eliminate the overlapping time and reduce overtime, idle time and the number crews needed to perform all the tasks of the company. This research presents a hybrid meta-heuristic known as Clustering Search (CS) to solve the problem. Instances found in the literature were used to evaluate the performance of the proposed method and the results were compared against other methods which have been studied to solve the CSP.

Keywords: Clustering Search; Crew Scheduling Problem; Metaheuristics.

LISTA DE FIGURAS

Figura 1- Representação do bloco de um veículo 13

Figura 2 - Representação de uma viagem 13

Figura 3 - Representação de um turno 14

Figura 4 - Modelagem do problema 21

Figura 5 - Fluxograma da CS. 22

Figura 6 - Algoritmo SA. 23

Figura 7 - Representação do cálculo da distância. 24

Figura 8 - Pseudocódigo da CS. 25

Figura 9 - Formulação das instâncias utilizadas. 26

LISTA DE TABELAS

Tabela 1 - Parâmetros finais da meta-heurística híbrida CS. 27

Tabela 2 - Parâmetros finais da meta-heurística SA. 27

Tabela 3 - Resultados obtidos para a instância com 25 tarefas. 28

Tabela 4 - Resultados obtidos para a instância com 50 tarefas. 28

Tabela 5 - Resultados obtidos para a instância com 100 tarefas. 29

Tabela 6 - Resultados obtidos para a instância com 250 tarefas. 29

Tabela 7 - Resultados obtidos para a instância com 500 tarefas. 30

Tabela 8 - Análise geral dos resultados obtidos pela CS. 30

Tabela 9 - Comparação dos resultados com a literatura (25, 50 e 100 tarefas). 31

Tabela 10 - Comparação dos resultados com a literatura (250 e 500 tarefas). 32

SUMÁRIO

1 INTRODUÇÃO 11

1.1 O PROBLEMA E SUA IMPORTÂNCIA 12

1.1.1 Descrição do Problema 12

1.1.2 Importância do Trabalho 15

1.2 OBJETIVOS 16

1.2.1 Objetivo Geral 16

1.2.2 Objetivos Específicos 16

2 REVISÃO DA LITERATURA 17

3 METODOLOGIA 19

3.1 MODELAGEM MATEMÁTICA 19

3.2 MODELAGEM COMPUTACIONAL 20

3.3 CLUSTERING SEARCH 21

3.3.1 Gerador de Soluções 23

3.3.2 Processo de Agrupamento 24

3.3.3 Analisador de Agrupamento 24

3.3.4 Busca Local 24

3.3.5 Perturbação 25

3.4 TESTES COMPUTACIONAIS 26

4 RESULTADOS OBTIDOS 27

5 CONCLUSÕES 33

6 REFERÊNCIAS BIBLIOGRÁFICAS 34

1 INTRODUÇÃO

Problemas de otimização têm por objetivo a minimização ou maximização de uma função por meio da escolha sistemática de valores de variáveis limitados por um conjunto de restrições. Os principais objetivos desses problemas são aperfeiçoar a produção, o processo, tempo e o mais importante, reduzir os custos.

Constantemente nos deparamos com vários problemas, como corte de madeira na fabricação de móveis, criação de horários escolares, escolha de menor rota para o deslocamento de um local para o outro, elaboração de mapas cartográficos, distribuição de tarefas no setor de transporte coletivo, etc. Visto que tais problemas permeiam as rotinas diárias das empresas de transporte coletivo, há a grande necessidade de otimizá-los em busca de boas soluções**.** Entretanto, muitos desses problemas são complexos e inviáveis de serem resolvidos manualmente.

Dentre os problemas citados, o transporte coletivo seja ele rodoviário, ferroviário, aéreo ou marítimo, é um setor essencial da economia para a sociedade. Além de empregar uma boa parte da população local em suas atividades, oferece serviços diários para sua locomoção (CALVI, 2005).

Esse setor passa por um grande desafio em relação à distribuição de escala de trabalho às suas tripulações, conhecido como Problema de Escalonamento de Tripulações (PET). Este é um problema permanentemente estudado, visto que os sistemas de transporte estão em contínua transformação e exigem uma gestão eficiente e confiável dos recursos disponíveis, visando sempre a qualidade do atendimento ao usuário e o valor gasto com as viagens (MAURI; LORENA, 2004).

Para resolução desses problemas podem ser utilizadas heurísticas e meta-heurísticas, que podem ser definidas como algoritmos que fornecem soluções sem um limite formal de qualidade e que são avaliadas empiricamente, ou seja, por meio de testes rotineiros até alcançar ou se aproximar de valores ótimos.

Na literatura, pode-se encontrar várias aplicações de heurísticas e meta-heurísticas utilizadas para tentar alcançar boas soluções para este tipo de problema. Meta-heurísticas como *Algoritmos Genéticos, Busca Tabu, Simulated Annealing,* entre outras, permitem incluir vários tipos de condições de trabalho apresentando

facilmente bons resultados (SHEN; KWAN, 2001; LORENÇO et al., 2001; MAURI, 2003).

Diante do exposto, neste trabalho é abordada uma meta-heurística híbrida conhecida como *Clustering Search* (CS), aplicada juntamente com a meta-heurística *Simulated Annealing* (SA), na busca por boas soluções que obedece às restrições impostas pelo PET.

1.1 O PROBLEMA E SUA IMPORTÂNCIA

1.1.1 Descrição do Problema

O Problema de Escalonamento de Tripulações (PET), também conhecido como *Crew Scheduling Problem* (CSP), apresentado por Wren e Rousseau (1993) ou *Bus Driver Scheduling Problem* (BDSP) mostrado por Lourenço et al. (2001), refere-se à atribuição de tarefas às tripulações de uma empresa de transporte, de tal forma que as viagens das diferentes linhas atendidas pela empresa sejam todas executadas com o menor custo possível (MAURI, 2005).

As tripulações são constituídas de um motorista e um cobrador, e as tarefas são referentes a um trecho de rota que pode ser representado pelo caminho de uma localidade A até B.

O PET consiste em distribuir ou escalonar tarefas às tripulações, de modo que garantam que todas as tarefas serão cobertas, além de atender um conjunto de restrições legais e estabelecidas pela empresa, visando obter uma combinação que traga um menor custo possível para realização das tarefas.

Normalmente, antes de ocorrer o escalonamento de tarefas para uma tripulação, é realizada a programação do veículo. Nesta, as viagens são agrupadas em blocos (Figura 1), sendo que cada bloco representa um conjunto de viagens que um veículo tem que realizar em um dia, geralmente começando e terminando na garagem. O conjunto de blocos forma uma escala para um ônibus (CALVI, 2005).

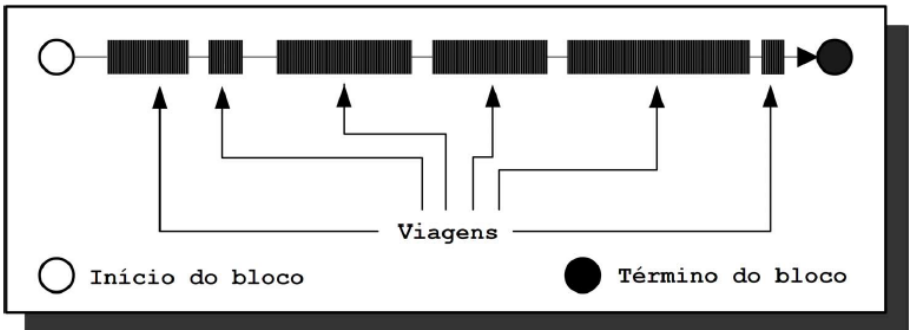

Figura 1- Representação do bloco de um veículo.
Fonte: Mauri e Lorena (2004).

Dentro de cada bloco são identificadas as oportunidades de troca (Figura 2), também conhecidas como *Relief Opportunities,* segundo Lourenço et al. (2001) ou *Relief Points,* segundo Desrochers e Soumis (1989). Uma oportunidade de troca é determinada como sendo os locais possíveis para que uma tripulação possa deixar um ônibus e outra assumir. Esses locais podem ser definidos como rodoviária, terminal, etc.

As viagens são organizadas em tarefas, a partir das quais são formados os turnos que serão executados por uma tripulação durante um dia de trabalho, como pode ser visto na Figura 2.

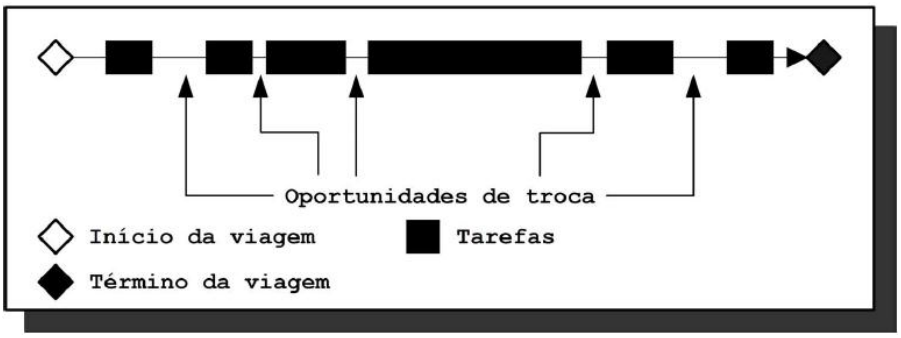

Figura 2 - Representação de uma viagem.
Fonte: Mauri e Lorena (2004).

Na formação dos turnos de trabalho devem ser levadas em consideração algumas restrições especificas das organizações. Essas restrições geralmente são combinadas de regras nacionais e locais, podendo ser obrigatórias ou não. A Figura 3 ilustra o turno de trabalho de uma tripulação.

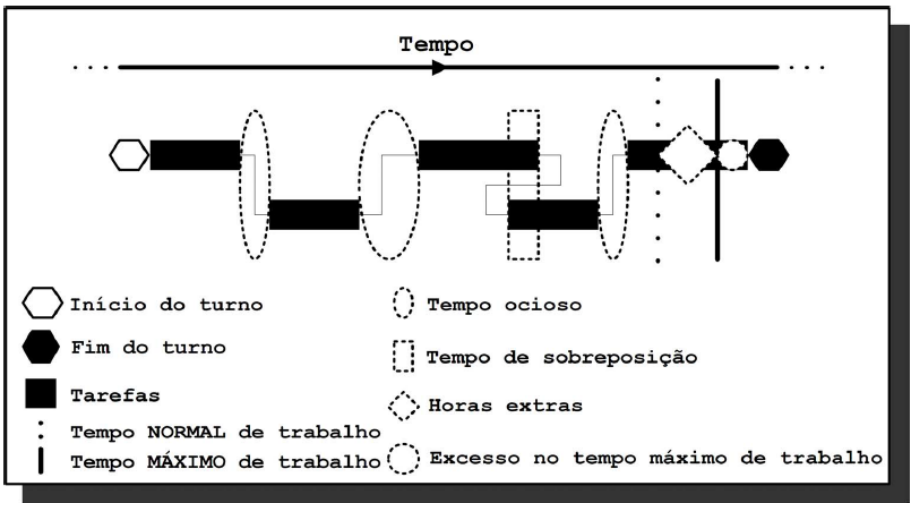

Figura 3 - Representação de um turno.
Fonte: Mauri e Lorena (2004).

Algumas restrições do problema são representadas na Figura 3. Essas restrições devem ser seguidas na hora da distribuição das tarefas, para garantir a viabilidade, a eficiência e a qualidade da solução obtida. Entre elas, se destacam o *tempo de sobreposição*, indicando que uma tripulação não pode realizar duas ou mais tarefas ao mesmo tempo. Esse tempo deve ser zero para que a solução seja viável. O *tempo ocioso* representa o tempo que uma tripulação fica parada esperando o inicio da próxima tarefa. É desejado que esse tempo seja reduzido. *Horas extras* indica o valor excedido, além do *tempo normal* de trabalho, e deve ser minimizado. *Tempo máximo* de trabalho indica o tempo máximo de um turno considerando as horas extras, e *tempo excessivo* indica o tempo que um turno passou do tempo máximo de trabalho.

Para a solução ser viável, foi necessário aplicar um peso sobre as restrições descritas no tópico anterior, sendo que o *tempo sobreposto* e o *tempo excessivo* foram multiplicados por um peso igual a 1000. Assim, soluções que possuírem *tempo sobreposto* e *tempo excessivo* diferentes de zero não serão consideradas, pois o valor do custo final para a realização de todas as tarefas será muito alto, sendo a solução descartada. Já o *tempo ocioso* e a *hora extra* foram multiplicados por um peso igual a 1, pois são valores que podem ocorrer, e não inviabilizam a solução.

Por fim, com os turnos gerados, busca-se produzir uma distribuição de modo que o custo para a realização desses turnos seja o menor possível. Deste modo, é

desejável realizar um escalonamento de turnos que cubram todas as tarefas formadas a partir dos blocos dos veículos, sem deixar de atender todas as regras e restrições apresentadas para o problema, visando o menor custo possível (MAURI, 2005). A redução desses custos pode ser alcançada gerando uma distribuição de modo que diminua a hora extra, o tempo ocioso e o número de tripulações necessárias para que a empresa realize todas as suas tarefas.

1.1.2 Importância do Trabalho

As empresas de transporte coletivo representam uma grande fatia da economia dos países e, além de serem geradoras de empregos, oferecem condução a milhões de pessoas. Entretanto, mesmo possuindo um bom potencial econômico, as empresas sofrem com os altos custos, principalmente de mão-de-obra que, segundo Calvi (2005), correspondem a aproximadamente 50% do custo operacional total.

A distribuição de tarefas para tripulações não é uma atividade simples, principalmente se realizada manualmente e para uma frota de tamanho consideravelmente grande, devido às várias combinações diferentes que podem ser formadas. Assim, com um programa computacional, a aplicação de meta-heurísticas pode ser de grande auxílio para a elaboração do plano de distribuição de tarefas, de maneira que atenda todas as restrições das empresas e restrições impostas sobre ela, resultando assim em uma grande economia de tempo na elaboração desses planos.

Com um escalonamento das tripulações otimizado, a empresa provavelmente terá um maior lucro em curto prazo, devido à redução do custo operacional, e ainda terá uma economia em longo prazo, pois respeitando as regras trabalhistas, evitaria reclamações de seus funcionários (CALVI, 2005).

Um bom escalonamento de tripulações traz benefícios não somente para as empresas, mas também para seus funcionários e clientes. Os funcionários terão suas jornadas bem divididas, considerando suas horas de descanso e também sua carga de horário máxima de trabalho, para que situações indesejadas não ocorram, como trabalho excessivo, por exemplo. Já os clientes serão beneficiados, pois com a redução do custo operacional na realização das tarefas, ao final pode-se ter um ganho significativo no custo total, o que pode levar a redução do valor das tarifas

para os usuários finais, justificando assim qualquer trabalho no sentido de minimizar os custos com a mão-de-obra (MAURI; LORENA, 2004).

1.2 OBJETIVOS

A seguir são descritos os objetivos geral e específicos deste trabalho.

1.2.1 Objetivo Geral

Resolver o Problema de Escalonamento de Tripulações (PET) com a aplicação da meta-heurística híbrida *Clustering Search* (CS) associada com a meta-heurística *Simulated Annealing* (SA).

1.2.2 Objetivos Específicos

a. Modelagem do PET para ser adaptado para a CS.
b. Implementação da CS.
c. Experimentos computacionais com o método proposto utilizando instâncias disponíveis na literatura.
d. Comparação dos resultados obtidos pelo método proposto com resultados encontrados na literatura.

2 REVISÃO DA LITERATURA

O Problema de Escalonamento de Tripulações (PET) vem sendo estudado permanentemente, por diferentes autores. Na literatura, existem diversas metodologias utilizadas para a resolução do PET. A seguir, são apresentados alguns dos principais trabalhos referentes ao problema.

Lourenço et al. (2001) apresentaram uma nova meta-heurística multiobjectivo, baseada na Busca Tabu e Algoritmos Genéticos, para resolução do problema PET em situações reais. Essa abordagem foi aplicada a um sistema de suporte a decisões (chamado GIST) para auxiliar o departamento de planejamento de companhias de transportes público e privado, ou também a autoridades de trânsito em seus gerenciamentos. Essa abordagem multiobjectivo foi aplicada com sucesso ao GIST, pois conseguiu alcançar soluções eficientes em um curto espaço de tempo.

Mauri (2003) propôs a utilização da meta-heurística *simulated annealing* para resolução do PET. Sua metodologia baseava-se em movimentos de troca, nos quais a tarefa de uma tripulação era trocada com a de outra tripulação, em busca de uma melhor combinação. Essa meta-heurística trouxe boa solução para o PET, conseguindo diminuir os custos com a realização das tarefas da empresa e ao mesmo tempo seguindo as restrições estabelecidas para o problema.

Mauri e Lorena (2004) apresentaram um Algoritmo de Treinamento Populacional (ATP) que é utilizado para gerar colunas para formação de um problema de particionamento de conjuntos, que é resolvido através da Programação Linear (PL). O ATP busca gerar colunas (tripulações) que tenham baixo custo e que possuam uma cobertura suficiente das tarefas.

Bassi et al. (2007) propuseram a aplicação da meta-heurística GRASP - *Greedy Randomized Adaptive Search Procedure* (FEO; RESENDE, 1995), com o mecanismo de Reconexão por Caminhos. Neste trabalho, foi comprovado que a Reconexão por Caminhos utilizada como instrumento de intensificação é mais eficiente do que quando utilizada como pós-otimização. Os resultados obtidos foram superiores aos outros métodos clássicos utilizados na resolução do problema.

As meta-heurísticas abordadas neste trabalho já foram aplicadas em outros problemas e obtiveram bons resultados, como pode ser visto nos trabalhos descritos a seguir.

Em Almeida e Senne (2010), a CS com SA foi utilizada na busca por agrupamentos, aplicados ao problema de localização de *hubs* com restrições de capacidades. Já Araújo e Mauri (2013) também propuseram a utilização da CS com SA para resolução do problema de rotação de culturas com restrições de adjacências. Foi obtida uma boa solução, alcançando resultados melhores do que outros métodos encontrados na literatura.

Costa et al. (2013) propuseram a aplicação das meta-heurísticas *Clustering Search* (CS) e *Simulated Annealing* (SA) para resolução de um problema semelhante ao PET, que é o problema de escalonamento de motoristas no transporte de madeira, aplicado em uma empresa localizada no Vale do Rio Doce - MG. As duas meta-heurísticas obtiveram bons resultados, sendo que a CS se destacou pelo seu baixo desvio e a SA pelo seu baixo tempo computacional, conseguindo uma redução de aproximadamente 1/3 das tripulações da empresa.

Rabello et al. (2013) utilizaram a CS para a resolução do problema de rotulação cartográfica de pontos, para o qual a CS também apresentou melhores soluções comparadas com outros trabalhos relacionados ao problema encontrados na literatura até então.

Por fim, mais detalhes sobre o PET e dos estudos a seu respeito podem ser encontrado em Ernst et al. (2004) e Wren (2004).

3 METODOLOGIA

A metodologia utilizada neste trabalho consiste na aplicação da meta-heurística híbrida *Clustering Search* (CS) utilizando o método *Simulated Annealing* (SA) como gerador de soluções para resolução do Problema de Escalonamento de Tripulações (PET).

3.1 MODELAGEM MATEMÁTICA

O Problema de Escalonamento de Tripulações (PET), como vários outros problemas de otimização, é um problema de minimização, mas especificamente voltado para a minimização dos custos em relação às tarefas realizadas pela empresa. No PET, deve-se distribuir todas as tarefas às tripulações disponíveis, sendo que cada tarefa só poderá ser executada por uma única tripulação. Assim, o custo de cada tripulação será avaliado de acordo com a seguinte expressão:

$$Min \sum_{j-1}^{n} c_j = ET_j + TS_j + HE_j + TO_j \tag{1}$$

Sendo que:

$$ET_j = max\left(0, \left[HT(t_{ult}) - HI\left(t_{pri}\right)\right] - TMT\right)$$

$$TS_j = \sum_{i=1}^{p-1} max(0, [HT(t_i) - HI(t_{i+1})])$$

$$HE_j = max\left(0, \left[HT(t_{ult}) - HI\left(t_{pri}\right)\right] - TNT\right)$$

$$TO_j = max\left(0, TNT - \left[HT(t_{ult}) - HI\left(t_{pri}\right)\right]\right) + \sum_{i=1}^{p-1} max(0, [HI(t_{i+1}) - HT(t_i)])$$

Em que:

- c_j - Representa o custo da tripulação *j*.
- *n* - Número de tripulação.
- ET_j – É o excesso no tempo máximo de trabalho da tripulação *j*.
- TS_j – É o tempo de sobreposição da tripulação *j*.
- HE_j – É o tempo que excede o tempo normal de trabalho da tripulação *j*.

- TO_j – É o tempo ocioso da tripulação *j.*
- *TMT* – É o tempo máximo de trabalho (8 horas tempo normal + 2 horas extras).
- *TNT* – É o número normal de trabalho (8 horas).
- *p* – É o número de tarefas do turno da tripulação *j*.
- $HT(t_i)$ – É o horário de termino da tarefa *i*.
- $HI(t_i)$ – É o horário de inicio da tarefa *i*.
- t_{pri} – É a primeira tarefa do turno da tripulação *j*.
- t_{ult} - É a última tarefa do turno da tripulação *j*.

O objetivo do PET pode ser descrito então como a minimização do somatório do custo de todas as tripulações, dado pela expressão (1). O custo de cada tripulação consiste em um somatório de todas as Horas Extras, Tempo Ocioso, Tempo Sobreposto e Tempo Excedente que cada tripulação terá ao realizar todas as tarefas que lhe foram destinadas.

3.2 MODELAGEM COMPUTACIONAL

A modelagem computacional do problema foi feita de forma a simplificar a sua implementação. Para isso, foram utilizadas três estruturas, sendo que a primeira armazena as informações de uma tripulação, que consiste de todos os atributos necessários para a execução do problema, além de um vetor com as tarefas de cada tripulação. A segunda estrutura é a solução, ela armazena os valores totais dos atributos e um vetor que armazena a estrutura de cada tripulação. Por fim, uma estrutura utilizada para execução da *Clustering Search* (CS).

Para uma melhor compreensão é apresentada na Figura 4 uma representação do funcionamento das estruturas. A estrutura solução possui um vetor de Tripulações, que por sua vez armazena uma estrutura que contém todas as informações da tripulação. Já a estrutura tripulação possui um vetor para as tarefas que uma tripulação realizará, na qual a partir delas são calculadas todas as variáveis necessárias para resolver o problema em estudo. Exemplo, a primeira posição da Estrutura Solução representa uma solução para o problema que possui 8 (oito) tripulações, onde a tripulação de identificação igual a 1 (um) está realizando as

tarefas 1 (um), 2 (dois) e 3 (três) das 25 (vinte e cinco) disponíveis que deverão ser distribuídas para o restante das tripulações.

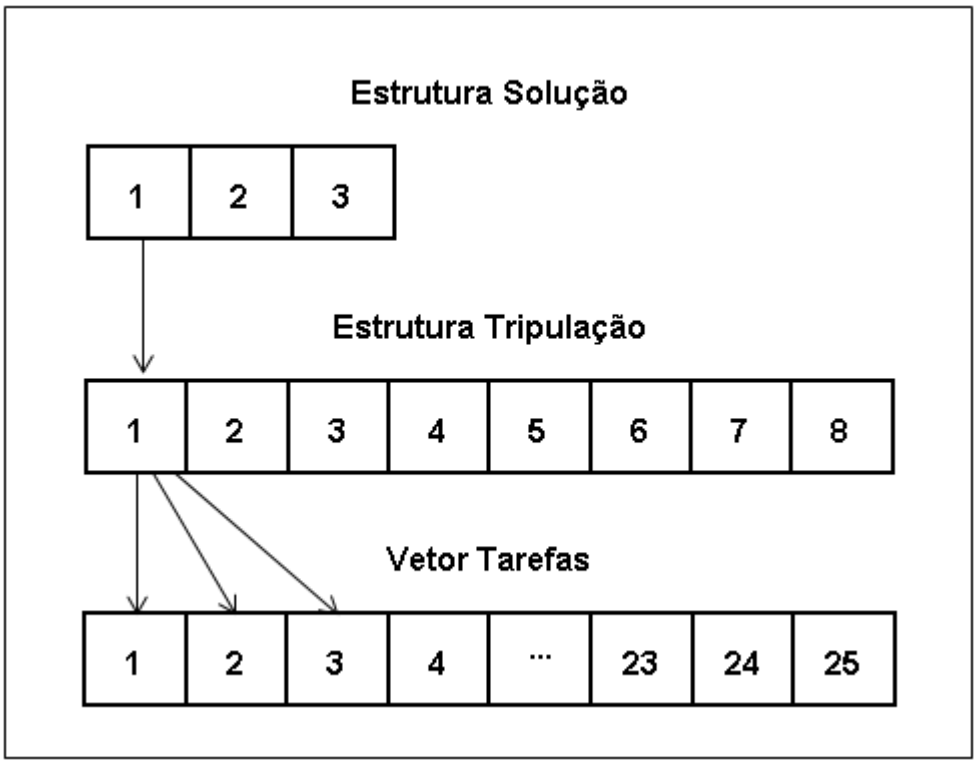

Figura 4 - Modelagem do problema.

As estruturas necessárias para o problema foram implementadas de maneira estática, buscando um menor custo computacional para resolução do PET, dispensando a necessidade de alocação e desalocação de memória durante execução do método.

3.3 CLUSTERING SEARCH

A *Clustering Search* – CS (OLIVEIRA; LORENA, 2007) surgiu após evoluções no algoritmo *Evolutionary Clustering Search* (ECS) proposto por Oliveira e Lorena (2004). No ECS, o processo de agrupamento para enquadrar os indivíduos promissores é feito simultaneamente com um algoritmo evolutivo. Já na CS, é utilizada uma meta-heurística qualquer no lugar do algoritmo evolutivo, como o GRASP, Busca Tabu, *Simulated Annealing* (SA), etc. (CHAVES; LORENA, 2010).

Segundo Chaves e Lorena (2010), a ideia principal da CS é procurar dividir o espaço de busca e localizar regiões promissoras por meio do enquadramento dessas

regiões em *clusters*, sendo a quantidade de *clusters* definida previamente como um parâmetro da CS.

Um *cluster* pode ser dividido em um conjunto (c_i,v_i,r_i), em que c_i, v_i e r_i representam respectivamente o centro, o volume e a ineficácia de busca relacionada ao *cluster i.* O centro c_i armazena uma solução que representa o *cluster i.* Inicialmente, essa solução pode ser obtida de maneira aleatória e vai sendo substituída conforme soluções mais promissoras são obtidas. O volume v_i representa o número de soluções presentes no *cluster i*, e r_i indica a quantidade de vezes que a busca local é aplicada no *cluster i* sem melhorar a solução. r_i é conhecido como índice de ineficácia do *cluster*. Para evitar que, regiões que já tenham sido suficientemente exploradas, continuem sendo consideradas pela busca local, é definida uma variável r_{max} com um número máximo de vezes que o r_i pode chegar.

A Figura 5 representa o fluxograma de execução da CS, no qual se destacam os seus quatro principais componentes, que são: 1 – uma meta-heurística geradora; 2 – um processo de agrupamento interativo; 3 – um analisador de agrupamento; e 4 – uma heurística de busca local.

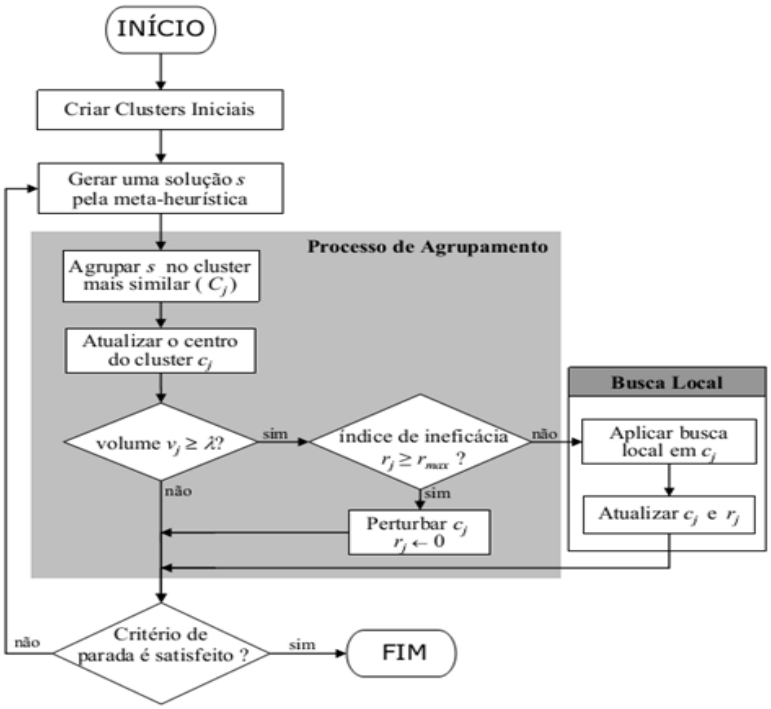

Figura 5 - Fluxograma da CS.
Fonte: Chaves e Lorena (2010).

3.3.1 Gerador de Soluções

A meta-heurística geradora se baseia na criação de uma grande quantidade de soluções diferentes que é fornecida à CS. A meta-heurística geradora utilizada neste trabalho é a *Simulated Annealing* (SA), pelo motivo de ter apresentado bons resultados quando aplicado em conjunto com a CS em outros problemas de otimização, como visto na revisão da literatura deste trabalho.

A *Simulated Annealing* (SA) foi originalmente proposta por Kirkpatrick et al. (1983) e refere-se a um método de busca local que aceita movimentos de piora para escapar de ótimos locais. Esse método começa a realizar suas buscas a partir de uma solução inicial aleatória. A SA se fundamenta em uma analogia com a termodinâmica, ao simular o resfriamento de um conjunto de átomos aquecidos. O algoritmo de busca trabalha em um laço de repetições, onde consideram uma temperatura *T*, em que *T* começa com um valor elevado, e em cada iterações geram-se soluções em temperaturas decrescentes, até que o equilíbrio de *T* seja alcançado. A temperatura é rebaixada até que entre no estágio de congelamento, no qual não se consegue mais melhoras naquela solução. A Figura 6 apresenta um algoritmo básico da SA.

```
1.  DADO (α, SAmax, T0, TC e S) FAÇA
2.  S* ← S;          {Melhor solução obtida até então}
3.  IterT ← 0;       {Número de iterações na temperatura T}
4.  T ← T0;          {Temperatura corrente}
5.  ENQUANTO (T > TC) FAÇA
6.     ENQUANTO (IterT < SAmax) FAÇA
7.          IterT ← IterT ++;      S' ← S;
8.          GERAR (um vizinho qualquer S' ∈ Ψ(S));
9.          SE (f(S') < f(S))
10.            S ← S';
11.            SE (f(S') < f(S*))    S* ← S';  FIM-SE;
12.         SENÃO
13.            TOMAR (x ∈ [0,1]);
14.            SE (x < e^(-Δ/T))    S ← S';    FIM-SE;
15.         FIM-SE;
16.    FIM-ENQUANTO;
17.    T ← α * T; IterT ← 0;
18. FIM-ENQUANTO;
19. S ← S*;
20. RETORNAR (S);
```

Figura 6 - Algoritmo SA.
Fonte: Mauri (2005).

3.3.2 Processo de Agrupamento

Após a CS receber uma solução, é iniciado o processo de agrupamento, o qual atribuirá uma solução *s* ao *cluster* mais próximo. Essa atribuição ocorre de acordo com a distância de Hamming H_i (HAMMING, 1950) que, neste trabalho, foi calculada com base na diferença entre os valores das variáveis das soluções *s* e s_i. Assim a soma dessas diferenças é o valor da distância de uma solução para outra (Figura 7).

Solução	Hora_Extra	Tempo_Excedente	Tempo_Ocioso	Tempo_Sobreposição
S	**1**	2	**3**	5
S_1	**6**	2	**1**	5

Distância = 7

Figura 7 - Representação do cálculo da distância.

3.3.3 Analisador de Agrupamento

Para a atualização do centro do *cluster*, são comparados os valores da nova solução com a que está atualmente no centro do *cluster* e, se a nova solução for melhor, o centro é atualizado com o seu valor, senão o valor do centro continua inalterado. Após essa etapa, é verificado o seu volume v_i e, caso v_i seja maior ou igual a γ, significa que o *cluster* é promissor e seu índice de ineficácia r_i deve ser analisado, sendo que se o r_i for maior ou igual r_{max} indica que a busca local não está melhorando o centro do *cluster*, daí é aplicada uma perturbação para evitar que buscas exaustivas fiquem sendo feitas em locais já explorados. Caso contrário, é realizada a busca local em seu centro, e sempre que melhores soluções forem encontradas o centro do *cluster* c_i é atualizado.

3.3.4 Busca Local

A busca local é aplicada sempre que o volume v_i de um *cluste*r alcança um limite γ, ou seja, o número de soluções assimiladas no *cluster* foi alta, então uma busca local é aplicada no centro deste *cluster* para intensificar a busca por melhores resultados nesse espaço.

A busca local implementada neste trabalho possui um simples funcionamento. Primeiramente, recebe por parâmetro uma solução *s* que está no centro do *cluster* e será utilizada como base para realizar buscas por combinações melhores.

O objetivo é testar todas as tarefas em todas as tripulações, buscando combinações que retornem soluções melhores. O processo de distribuição ocorre de forma que uma tarefa é testada em todas as tripulações e a melhor solução obtida com os testes é armazenada. Após todas as tarefas de uma tripulação serem testadas em todas as outras tripulações e a sua solução continuar melhorando, o procedimento é feito novamente com as tarefas das próximas tripulações.

3.3.5 Perturbação

O processo de perturbação é bem simples, basta apenas modificar de alguma maneira a solução corrente. Neste trabalho, a perturbação é feita realizando um movimento de troca, que consiste de trocar de forma aleatória uma tarefa de uma tripulação com a tarefa de outra tripulação. Outro movimento de troca implementado neste trabalho é a troca de duas tarefas de uma só vez entre duas tripulações.

A Figura 8 apresenta o pseudocódigo da CS utilizada neste trabalho.

```
1.  Criar γ novas soluções (clusters)
2.  v_i ← 0; e r_i ← 0 ∀ i = 1,...,γ; s ← solução inicial;  s* ← s;  T ← T_0;
3.  ENQUANTO T > T_C FAÇA
4.     iter ← 0
5.     ENQUANTO iter < SA_max FAÇA
6.        iter ← iter ++;   s' ← solução vizinha de s;
7.        SE f(s') > f(s) ENTÃO
8.           s ← s';
9.        SENÃO
10.          Com probabilidade e^(-(f(s)-f(s'))/T) s ← s';
11.       FIM-SE
12.    FIM-ENQUANTO
13.    T ← αT;     i ← arg min_{i∈{1,...,γ}} {H_i};
14.    v_i ← v_i + 1;        c_i ← melhor(s,c_i);
15.    SE v_i = λ ENTÃO
16.       v_i ← 0 ;   s ← busca local(c_i);
17.       SE f(s) = f(c_i) ENTÃO
18.          r_i ← r_i + 1;
19.          SE r_i = r_max ENTÃO
20.             r_i ← 0;    c_i ← N(c_i);
21.          FIM-SE
22.       SENÃO
23.          r_i ← 0;
24.       FIM-SE
25.    FIM-SE
26.    s* ← max(s*,c_i);
27. FIM-ENQUANTO
28. Retornar s*;
```

Figura 8 - Pseudocódigo da CS.

3.4 TESTES COMPUTACIONAIS

Para verificação do desempenho do método proposto, foi utilizado um conjunto de instâncias baseadas em problemas reais de uma empresa de transporte público brasileira. As instâncias foram propostas por Mauri (2005), e representam 25, 50, 100, 250, 500 tarefas, que são compostas por um horário de inicio e de fim, selecionados aleatoriamente de um problema real. A Figura 9 apresenta a formulação de cada instância.

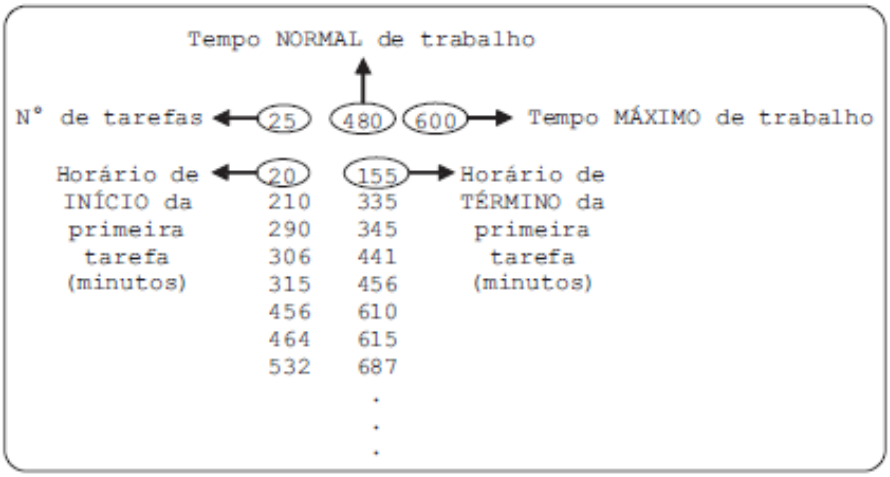

Figura 9 - Formulação das instâncias utilizadas.
Fonte: Mauri (2005).

Para cada instância do problema, a CS foi executada 10 vezes, sendo analisados os seguintes itens para cada execução do problema.

- **Tripulação Final:** representa o número necessário de tripulações para realizar todas as tarefas da instância testada;
- **Função Objetivo:** valor da função objetivo;
- **Desvio Médio:** é calculado pela fórmula

  $$D = (((ML - ME) / ML) * 100)$$

  Em que *ML* é o valor da função objetivo da melhor solução encontrada na execução dos testes de cada instância e *ME* é o valor médio das execuções;
- **Tempo:** representa o tempo total, em segundos, para a execução do método para resolução de uma instância do problema.

4 RESULTADOS OBTIDOS

A seguir, são apresentados os resultados obtidos por meio da execução da meta-heurística *Clustering Search* (CS) proposta. Para comparação, foram utilizados os resultados obtidos pelos métodos *simulated annealing* (SA) e ATP/PL propostos por Mauri (2005).

Os parâmetros utilizados neste trabalho para a execução da CS e SA foram obtidos através de testes, e são apresentados nas Tabelas 1 e 2 respectivamente.

Tabela 1 - Parâmetros finais da meta-heurística híbrida CS.

Parâmetro	Descrição	CS
λ	Quantidade máxima de clusters	10
γ	Volume máximo para aplicação da busca local	7
r_{max}	Limite para aplicação da perturbação	4

Tabela 2 - Parâmetros finais da meta-heurística SA.

Parâmetro	Descrição	SA
T_o	Temperatura inicial	1000
T_c	Temperatura de congelamento	0,01
SA_{max}	Número máximo de iterações	1000
α	Taxa de resfriamento	0.975

Com os parâmetros descritos nas Tabelas 1 e 2, a CS, utilizando com o SA como geradora de soluções, foi aplicada 10 (dez) vezes para cada instância. Os resultados são apresentados nas Tabelas 3, 4, 5, 6 e 7, para as instâncias com 25, 50, 100, 250 e 500 tarefas, respectivamente. As primeiras instâncias, com 25, 50 e 100 tarefas, foram executadas por 600 segundos em cada uma das 10 (dez) vezes. Já a instância de 250 executou por 7200 segundos e a instância com 500 tarefas por 13000 segundos. Os melhores valores encontrados entre as 10 vezes que cada método foi executado para cada instância, estão em destaque em cada tabela. Já os tempos apresentados nessas tabelas indicam o momento em que as melhores soluções foram encontradas dentro do limite descrito anteriormente.

Tabela 3 - Resultados obtidos para a instância com 25 tarefas.

Execução	Número de Tripulações	Horas Extras (min.)	Tempo Ocioso (min.)	Custo Total da Solução	Tempo (seg.)
1	12	15	2356	2371	1,19
2	12	15	2356	2371	4,28
3	12	15	2356	2371	3,85
4	12	15	2356	2371	0,92
5	**12**	**15**	**2356**	**2371**	**0,39**
6	12	15	2356	2371	0,55
7	12	15	2356	2371	1,90
8	12	15	2356	2371	1,28
9	12	15	2356	2371	0,67
10	12	15	2356	2371	0,69
Média	**12**	**15**	**2356**	**2371**	**1,57**

Tabela 4 - Resultados obtidos para a instância com 50 tarefas.

Execução	Número de Tripulações	Horas Extras (min.)	Tempo Ocioso (min.)	Custo Total da Solução	Tempo (seg.)
1	20	0	2600	2600	14,26
2	20	0	2600	2600	13,84
3	**20**	**0**	**2600**	**2600**	**7,27**
4	20	0	2600	2600	14,29
5	20	0	2600	2600	13,99
6	20	0	2600	2600	13,56
7	20	0	2600	2600	10,48
8	20	0	2600	2600	14,20
9	20	0	2600	2600	13,98
10	20	0	2600	2600	13,79
Média	**20**	**0**	**2600**	**2600**	**12,96**

Tabela 5 - Resultados obtidos para a instância com 100 tarefas.

Execução	Número de Tripulações	Horas Extras (min.)	Tempo Ocioso (min.)	Custo Total da Solução	Tempo (seg.)
1	40	0	7395	7395	9,45
2	40	0	7395	7395	9,64
3	40	0	7395	7395	7,99
4	**40**	**0**	**7383**	**7383**	**8,08**
5	40	0	7395	7395	0,20
6	40	0	7395	7395	11,93
7	40	0	7395	7395	9,66
8	40	0	7395	7395	8,03
9	40	0	7395	7395	10,81
10	40	0	7395	7395	13,46
Média	**40**	**0**	**7393**	**7393**	**8,92**

Tabela 6 - Resultados obtidos para a instância com 250 tarefas.

Execução	Número de Tripulações	Horas Extras (min.)	Tempo Ocioso (min.)	Custo Total da Solução	Tempo (seg.)
1	89	525	11013	11538	35,04
2	**89**	**351**	**10839**	**11190**	**638,18**
3	90	257	11225	11482	2166,64
4	89	534	11022	11556	18,47
5	90	294	11262	11556	0,42
6	90	276	11244	11520	335,28
7	89	418	10906	11324	1192,10
8	88	684	10692	11376	479,02
9	89	469	10957	11426	7124,03
10	89	468	10956	11424	18,19
Média	**89,2**	**427,6**	**11011**	**11439**	**1200,73**

Tabela 7 - Resultados obtidos para a instância com 500 tarefas.

Execução	Número de Tripulações	Horas Extras (min.)	Tempo Ocioso (min.)	Custo Total da Solução	Tempo (seg.)
1	175	156	21070	21226	0,67
2	175	295	21209	21504	41,09
3	176	109	21503	21612	119,81
4	176	95	21489	21584	44,23
5	175	216	21130	21346	1,11
6	**174**	**151**	**20585**	**20736**	**0,42**
7	175	193	21107	21300	39,66
8	176	155	21549	21704	0,76
9	175	242	21156	21398	40,58
10	176	169	21563	21732	78,89
Média	**175,3**	**178,1**	**21236**	**21414**	**36,72**

A Tabela 8 apresenta uma análise geral dos resultados obtidos com a CS para as instâncias deste trabalho. Nessa tabela, são apresentados o número de tripulações obtido para realizar todas as tarefas das instâncias, o desvio médio, o melhor valor da FO entre as 10 FO obtidas para cada instância, o valor médio da FO, o tempo médio de execução total para cada instância e o tempo médio para encontra a melhor solução.

Tabela 8 - Análise geral dos resultados obtidos pela CS.

Número de Tarefas	Número de Tripulações	Desvio	Melhor FO	FO Média	Tempo Médio (seg.)	Tempo Melhor (seg.)
25	12	0%	2371	2371	600	1,57
50	20	0%	2600	2600	600	12,96
100	40	0,13%	7383	7393	600	8,92
250	89	2,22%	11190	11439	7200	1200,73
500	174	3,16%	20736	21414	13000	36,72

Para as instâncias com 25, 50, 100, 250 e 500 tarefas, a SA proposta por Mauri (2005), encontrou um tempo médio de processamento para cada instância de 1,98,

95,79, 67,19, 434,80, 12132,34 segundos respectivamente. O ATP/PL, também proposto por Mauri (2005), obteve 0,30, 41,64, 4,98, 229,68, 6797,40 segundos respectivamente.

Na execução da CS proposta neste trabalho para as instâncias com 25 e 50 tarefas, a meta-heurística encontrou soluções idênticas às encontradas por Mauri (2005) com a aplicação da SA e do ATP/PL. A CS obteve um desvio de 0% para essas instâncias, e também obteve o tempo médio de processamento igual a 1,57 e 12,96 segundos respectivamente, conseguindo um tempo de processamento para solucionar o problema com 50 tarefas, menor do que o já encontrado até então. Para a instância com 25 tarefas, a CS obteve um tempo melhor que o SA, porém foi inferior ao resultado encontrado pelo ATP/PL. Já para a instância com 100 tarefas, a CS obteve um tempo de processamento de 8,92 segundos, e um desvio de 0,13%, conseguindo a melhor solução encontrada entre os 3 (três) métodos, mesmo com o tempo para encontrar a solução pior que o ATP/PL.

Para as instâncias com 250 e 500 tarefas, a CS proposta não conseguiu alcançar os valores das soluções obtidas pelos métodos SA e ATP/PL, obtendo um desvio de 2,22% e 3,16% respectivamente.

As Tabelas 9 e 10 apresentam uma comparação dos resultados obtidos nesse trabalho com os resultados encontrados na literatura. Os melhores resultados entre os três métodos para cada instância estão destacados.

Tabela 9 - Comparação dos resultados com a literatura (25, 50 e 100 tarefas).

Métodos de solução	**Tarefas**					
	25		**50**		**100**	
	FO	**Tempo**	**FO**	**Tempo**	**FO**	**Tempo**
Clustering Search (CS)	2371	1,57	**2600**	**12,96**	**7383**	8,92
Simulated Annealing (MAURI, 2005)	2371	1,98	2614	95,79	7395	67,19
ATP/PL (MAURI, 2005)	2371	**0,30**	2600	41,64	7395	**4,98**

Tabela 10 - Comparação dos resultados com a literatura (250 e 500 tarefas).

Métodos de solução	**Tarefas**			
	250		**500**	
	FO	**Tempo**	**FO**	**Tempo**
Clustering Search (CS)	11190	1200,73	20736	36,72
Simulated Annealing (MAURI, 2005)	9902	434,80	12750	12132,34
ATP/PL (MAURI, 2005)	**9244**	**229,68**	**12528**	**6797,40**

A partir da análise das tabelas anteriores, observa-se que, para as instâncias com 25, 50 e 100 tarefas, os resultados obtidos pela CS foram iguais ou melhores aos resultados encontrados pelo SA, obtendo uma melhora em relação ao tempo médio de execução de 20,70%, 86,47% e 86,7% respectivamente, e também uma melhora na solução final encontrada. Já realizando uma comparação dessas mesmas instâncias com o método ATP/PL, a CS obteve uma melhor solução em relação ao custo para a instância com 100 tarefas e um melhor tempo de processamento para a instância com 50 tarefas, reduzindo 68,87% no tempo necessário para encontrar uma solução para o PET.

Já nas instâncias com 250 e 500 tarefas, a CS não conseguiu igualar aos melhores resultados obtidos até então, sendo que o método APT/PL foi o que obteve as melhores soluções para as instâncias com 250 e 500.

A CS proposta foi implementada na linguagem C++ e os experimentos computacionais foram realizados em um computador Dell XPS8500, com processador Intel(R) Core^{TM} i5-3450 CPU 3.10GHz e com 8,00 GB de memória.

Os resultados apresentados por Mauri (2005) para a SA e o ATP/PL, foram realizados utilizando um *laptop Toshiba A10 S127* com processador Intel Celeron de 2.0 GHz e 256MB de memória RAM. Sua implementação foi desenvolvida na linguagem C++ e o desvio entre as execuções das soluções não foi apresentado.

5 CONCLUSÕES

Este trabalho apresentou uma nova alternativa ainda não explorada para resolução do Problema de Escalonamento de Tripulações (PET).

Para resolução do PET, foi implementada a meta-heurística híbrida *Clustering Search* (CS) utilizando a *Simulated Annealing* (SA) como sua meta-heurística geradora.

A CS obteve bons resultados para as instâncias com 25, 50 e 100 tarefas. Já para as instâncias com 250 e 500 tarefas a SA apresentou resultados inferiores aos obtidos pelo SA e pelo ATP/PL propostos por Mauri (2005). Porém, para essas instâncias que os resultados não foram satisfatórios, as soluções poderiam ser melhoradas utilizando uma nova estratégia para a implementação da busca local, para deixá-la mais eficiente e rápida.

Por fim, é possível concluir que a meta-heurística proposta apresentou uma boa alternativa para auxiliar empresas de transporte coletivo na distribuição de tarefas a suas tripulações.

6 REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, G. W. E SENNE, E. L. F. Meta-heurística hibrida com busca por agrupamento Aplicado ao problema de localização de hubs com Restrições de capacidade. *Anais do XLI SBPO - Simpósio Brasileiro de Pesquisa Operacional*, 2010.

ARAÚJO, D. L.; MAURI, G. R. Clustering search para resolução de um problema de rotação de culturas com restrições de adjacências. *Anais do XLI SBPO - Simpósio Brasileiro de Pesquisa Operacional,* 2013.

BASSI, H. V.; MARINHO, E. H.; SOUZA, M. J. F.; Silva, G. P. Heurísticas GRASP e Reconexão por Caminhos aplicadas ao problema de escalonamento de tripulações. *Anais do XXXIX SBPO - Simpósio Brasileiro de Pesquisa Operacional*, 2007.

CALVI, R. Um algoritmo para o problema de escalonamento de tripulação em empresas de ônibus*.* Dissertação de mestrado - Universidade Estadual de Maringá (UEM), Maringá, 2005.

CHAVES, A. A.; LORENA, L. A. N. Clustering search algorithm for the capacitated centered clustering problem. *Computers & Operations Research*, v. 37, p. 552-558, 2010.

COSTA, M. F.; FIEDLER, N. C.; MAURI, G. R. Clustering search e simulated annealing para resolução do problema de escalonamento de motoristas no transporte de madeira. *Scientia Forestalis*, v. 41, n. 99, p. 299-305, 2013.

DESROCHERS, M.; SOUMIS, F. A Column Generation Approach to the Urban Transit Crew Scheduling Problem. *Transportation Science*, v. 23, p. 1–13, 1989.

ERNST, A.; JIANG, H.; KRISHNAMOORTHY, M.; OWENS, B.; SIER, D. An annotated bibliography of personnel scheduling and rostering. In: JIANG, H.; KRISHNAMOORTHY, M.; SIER, D. (eds.) Annals of operations research. Netherlands: *Kluwer Academic Publishers*, p. 21–144, 2004.

FEO, T. A.; RESENDE, M. G. C. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6, p.109-133, 1995.

HAMMING, R. W. Error detecting and error correcting codes. *Bell System Technical Journal,* v. 26, p. 147-160, 1950.

KIRKPATRICK, S.; GELLAT, D. C.; VECCHI, M. P. Optimization by simulated annealing. *Science*, v. 220, p. 671–680, May 1983.

LORENÇO, H. R.; PAIXÃO, J. P. P.; PORTUGAL, R. Multiobjective metaheuristics for the bus-driver scheduling problem. *Transportation Science*, 35, p. 331-343, 2001.

MAURI, G. R. Resolução do problema de programação de tripulações de um sistema de transporte público via *simulated annealing*. Relatório técnico - Departamento de Ciência da Computação (UFOP), Ouro Preto, 2003.

MAURI, G. R.; LORENA, L. A. N. Método interativo para resolução do problema de escalonamento de tripulações. *Anais do XXXVI SBPO - Simpósio Brasileiro de Pesquisa Operacional*, 2004.

MAURI, G. R. Novas heurísticas para o problema de escalonamento de tripulações. Dissertação de mestrado - Instituto Nacional de Pesquisa Espacial (INPE), São José dos Campos, 2005.

OLIVEIRA, A. C. M.; LORENA, L. A. N. Detecting promising areas by evolutionary clustering search. In Advances in Artificial Intelligence SBIA 2004. *Lecture Notes in Artificial Intelligence*, 3171, 385-394, 2004.

OLIVEIRA, A. C. M.; LORENA, L. A. N. Hybrid evolutionary algorithms and clustering search. *Hybrid Evolutionary Algorithms - Studies in Computational Intelligence,* 75, 77-99, 2007.

RABELLO, R. L.; MAURI, G. R.; RIBEIRO, G. M.; LORENA, L. A. N. Meta-heurística Clustering search aplicada ao problema de Rotulação cartográfica de pontos. *Anais do XLV SBPO - Simpósio Brasileiro de Pesquisa Operacional*, 2013.

SHEN, Y.; KWAN, R. S. K. Tabu search for driver scheduling. In: Voß, S.; Daduna, J. (eds.) Computer-aided scheduling of public transport. Berlin: *Springer-Velag*, p.121-136, 2001.

WREN, A. Scheduling vehicles and their drivers: forty years' experience. In: *International Conference on Computer-Aided Scheduling of Public Transport(CASPT)*, 9., 2004, San Diego.

WREN, A.; ROUSSEAU, J. M. Bus Driver Scheduling - An Overview. *6th International Workshop on Computer Aided Scheduling of Public Transport.* 1993.