

Trabalho Prático 2

Algoritmo de Colônia de Formigas para resolver o problema da p-Mediana com restrições de capacidade

Bernardo de Almeida Abreu¹

¹Departamento de Ciência da Computação – Universidade Federal de Minas Gerais
Disciplina: DCC191 - Computação Natural 2017/2

bernardoabreu@dcc.ufmg.br

1. Introdução

O problema da p-mediana é um problema de localização que consiste em decidir onde localizar P centros em uma rede, composta por vértices e arestas, de forma a minimizar a soma de todas as distâncias de cada vértice ao centro mais próximo. Ou seja, dado um grafo completo deve-se determinar P vértices, chamados centros, que se conectam aos outros vértices por arestas com peso, de forma que cada vértice do grafo esteja conectado a um e exatamente um centro e a soma dos pesos das arestas utilizadas seja mínima.

Nesse trabalho, é tratado o problema da p-mediana com restrição de capacidade. A restrição de capacidade faz com que, cada vértice esteja associado a uma capacidade e a uma demanda. Dessa forma, ao se escolher um centro, sua capacidade deve ser maior ou igual a soma das demandas de todos os vértices conectados a esse centro.

O problema da p-mediana é um problema de otimização combinatória NP-difícil. O algoritmo de colônia de formigas será usado para resolver esse problema nesse trabalho.

2. Modelo

O problema possui algumas restrições que devem ser atendidas.

- Cada vértice deve ser atendido por somente uma mediana ou ser uma mediana.
- Existem exatamente P medianas.
- A soma das demandas do vértices alocados para uma mediana não deve ser maior do que a capacidade da mediana.

A modelagem do problema segue o que foi feito no trabalho **Max Min Ant System and Capacitated p-Medians: Extensions and Improved Solutions** [de França et al. 2005]. Dessa forma o algoritmo de colônia de formigas é utilizado para escolher as P medianas da solução entre os pontos disponíveis. A distribuição dos pontos restantes, aqui chamados de clientes, é feita por meio de uma heurística.

O programa funciona da seguinte forma:

```

ant_colony():
    while it < max_it:
        for ant < num_ants:
            build_solution();
            eval_solution();
        end
        update_pheromone();
    end
end

```

Para cada formiga uma solução é construída na função *build_solution()* e em seguida avaliada. Após terem sido criadas soluções para cada formiga, as trilhas de feromônio são atualizadas.

2.1. Construção de uma solução

Uma solução é constituída por P medianas e $N - P$ pontos, onde N é o número total de pontos, alocados para essas medianas. O pseudocódigo para construção de uma solução se encontra a seguir:

```

build_solution():
    for size of medians < p_medians:
        medians <- probabilistic_transition(non_selected_points);
    end
    assign_clients_to_medians();
end

```

2.2. Regra de transição

As P medianas são alocadas seguindo as regras de transição do algoritmo de colônia de formigas. Cada formiga k escolhe um novo nó para ser mediana por iteração t , seguindo a probabilidade descrita pela regra 1:

$$p_i^k(t) = \begin{cases} \frac{[\tau_i(t)]^\alpha \cdot [\eta_i]^\beta}{\sum_{l \in J^k} [\tau_l(t)]^\alpha \cdot [\eta_l]^\beta} & \text{se } i \in J^k \\ 0 & \text{caso contrário} \end{cases} \quad (1)$$

Cada ponto i ainda não escolhido para ser uma mediana, representado por J^k , possui uma probabilidade de se tornar uma mediana que depende do feromônio depositado nesse ponto, $\tau_i(t)$, e do valor dado pela função heurística η_i nesse ponto. Os pontos já escolhidos não podem ser escolhidos novamente, logo a sua probabilidade de escolha é zero.

A função η foi definida como uma função gulosa cujo objetivo é obter a qualidade de se escolher um ponto como uma mediana da solução. Para um dado ponto i a ser adicionado na solução, é calculada sua densidade, ou seja, a quantidade de pontos que poderiam ser atribuídos a essa mediana sem ultrapassar sua capacidade dividida pela soma total das distâncias desses pontos a mediana. Os pontos adicionados a uma mediana não levam em consideração os pontos adicionados para outros pontos, logo, o mesmo ponto pode ser adicionado para mais de uma mediana ao se calcular essa função.

2.2.1. Atribuição de clientes para medianas

Os pontos não escolhidos como medianas devem ser atribuídos para as medianas escolhidas. Esse é um Problema Generalizado de Atribuição, que é resolvido por meio de uma heurística construtiva adotada por [de França et al. 2005]. Essa heurística aloca cada cliente para a primeira mediana possível, respeitando os limites de capacidade da mesma, tentando as medianas em ordem de distância para o ponto que se tenta alocar, da mais próxima para a mais distante.

2.3. Atualização do feromônio

Existe um valor de feromônio para cada ponto no problema. É definido um valor mínimo de 0.001 e máximo de 0.999 para esses valores e eles são inicializados com 0.5. A regra de atualização do feromônio é dada a seguir:

$$\tau_i = \tau_i + \rho(\Delta\tau_i - \tau_i) \quad (2)$$

O valor de $\Delta\tau_i$ corresponde ao incremento do feromônio e o valor de ρ a taxa de evaporação. A regra de atualização que determina $\Delta\tau_i$ é dada por:

$$\Delta\tau_i = \begin{cases} 1 - \frac{l_{best} - g_{best}}{l_{worst} - l_{best}} & \text{se } i \in g_{best}, l_{best} \\ 0 & \text{caso contrário} \end{cases} \quad (3)$$

O valor l_{best} é o valor da melhor solução da iteração atual, l_{worst} o valor da pior solução da iteração atual e g_{best} o valor da melhor solução desde o início do programa. O valor de $\Delta\tau_i$ é adicionado somente as medianas que estão presentes nas soluções de l_{best} e g_{best} , nos outros pontos seu valor é zero.

Após atualizar os valores dos feromônios e garantir que eles fiquem dentro do intervalo [0.001, 0.999], é feito um controle de estagnação. Esse controle é feito da seguinte forma, se a soma de todos os feromônios for igual a soma dos feromônios na situação em que P pontos possuem feromônio máximo e $N - P$ possuem feromônio mínimo, como mostrado na equação 4, ou seja, a situação em que a solução está estagnada em P pontos, os valores de feromônio voltam para o valor inicial.

$$\sum_i \tau_i = p.\tau_{max} + (n - p).\tau_{min} \quad (4)$$

3. Execução

O programa do trabalho foi implementado utilizando a linguagem *Python 3.5* e o pacote *numpy* é necessário para a execução. O nome do arquivo principal é *main.py*.

Um exemplo de comando de execução:

```
./main.py --ants 10 --iterations 100 --alpha 3 --beta 1 --file ../dataset/SJC1.dat
```

O parâmetro *file* é obrigatório e corresponde ao arquivos de entrada. Os outros parâmetros são opcionais e possuem o propósito de controlar a execução do programa.

file Arquivo de entrada.

ants Número de formigas (Valor padrão: 10).

iterations Número de iterações (Valor padrão: 100).

alpha Alfa. Influência do feromônio (Valor padrão: 3).

beta Beta. Influência da função heurística (Valor padrão: 1).

decay Taxa de evaporação do feromônio (Valor padrão: 0.1).

seed Semente para geração de números aleatórios (Valor padrão: Nenhuma).

stats Arquivo para salvar as estatísticas (Valor padrão: Nenhum).

Quando se escolhe um nome de arquivo para salvar as estatísticas, é criado um arquivo para cada estatística a ser salva. Caso o arquivo já exista, os novos dados serão inseridos ao final do arquivo, sem apagá-lo. Para cada métrica, é inserida uma única linha, com os valores das métricas para cada iteração, separadas por vírgulas.

4. Experimentos

O experimentos foram realizados em um computador com as seguintes configurações:

Processador: Intel Core i7 4^a geração.

Memória: 16GB DDR3 166MHz

Sistema Operacional: Ubuntu 16.04

4.1. Metodologia

Foram realizados diversos experimentos de forma a encontrar os melhores parâmetros para a execução do programa. Para cada teste, foi definido um parâmetro a ser variado e todos os outros foram fixados em um valor. Para cada variação desse parâmetro, o programa foi executado 30 vezes, cada uma com uma *seed* diferente para a geração dos números aleatórios. Ao término de um teste, um valor de parâmetro foi definido e passou a ser o padrão nos testes seguintes. A *seed* usada para cada execução foi o número entre 1 e 30 relativo aquela execução. Os gráficos são construídos a partir da média dos valores da métrica analisada para as 30 *seeds*.

Os testes foram realizados com três bases de dados. As informação sobre essas bases são dadas na tabela 1

Base	Pontos	Medianas	Valor Ótimo
SJC1.dat	100	10	17246,53
SJC2.dat	200	15	33225,88
SJC3b.dat	300	30	40635,80

Table 1. Bases de dados

4.2. Testes

4.2.1. Variação do número de iterações

O objetivo do primeiro teste realizado foi analisar as implicações derivadas das mudanças no número de iterações e, dessa forma, definir um valor para o parâmetro. Os parâmetros de execução foram os seguintes:

Número de Formigas: 10

Alfa: 3

Beta: 1

Taxa de evaporação: 0.1

Foram rodadas 500 iterações para cada base de teste. É possível observar nas figuras 1, 2 e 3 que há uma grande melhora da solução nas primeiras 50 iterações, e após isso o resultado melhora mais gradativamente. Nota-se que uma maior quantidade de iterações influencia em uma obtenção de um melhor resultado final, porém essa influência não é muito grande. Apesar disso, como os resultados mostraram-se melhores para mais iterações, foi fixado o número de iterações como 500 para os testes posteriores.

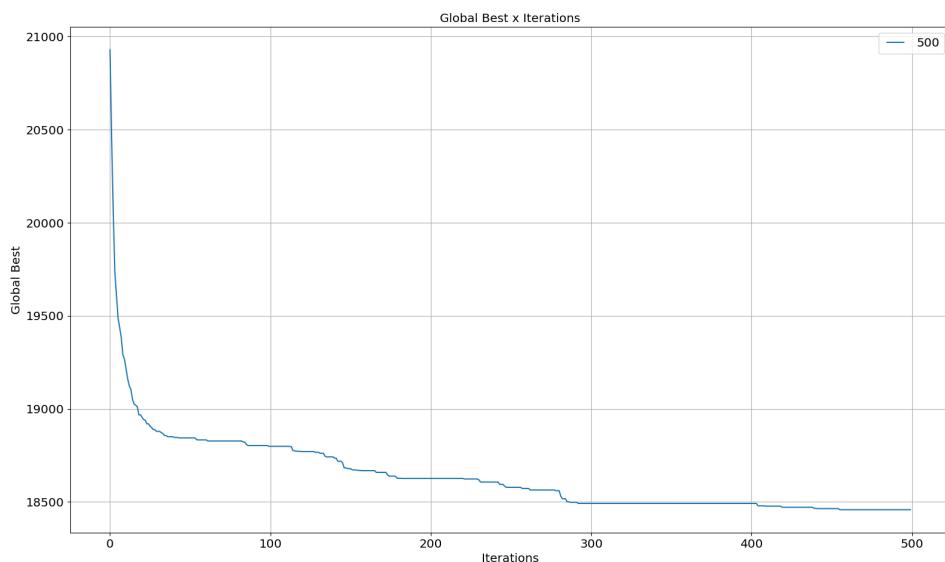


Figure 1. Melhor resultado global para SJC1.dat

4.2.2. Variação do número de formigas

O objetivo desse teste realizado foi analisar as implicações derivadas das mudanças no número de formigas. Os parâmetros de execução foram os seguintes:

Número de Iterações: 500

Alfa: 3

Beta: 1

Taxa de evaporação: 0.1

Os números de formigas escolhidas foram 10, 30, 60 e 90, baseada no valor de pontos e medianas da primeira base de testes. Esperava-se assim analisar o resultado para todas as bases e, caso necessário, executar mais testes para as bases maiores com mais formigas.

A figuras 4 e 5 mostram que tanto a melhor solução global quanto a melhor solução local melhoraram muito quando se aumenta o número de formigas de 10 para 30,

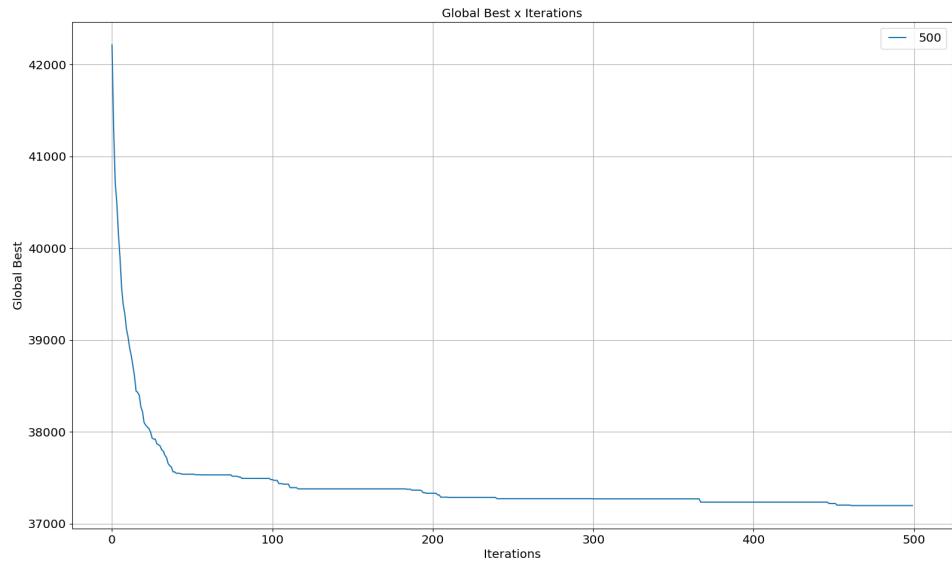


Figure 2. Melhor resultado global para *SJC2.dat*

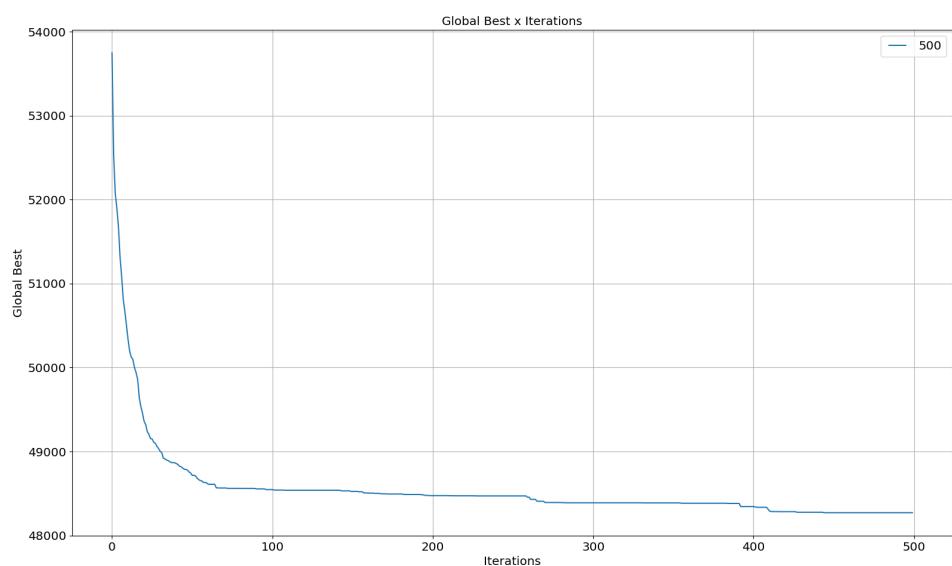


Figure 3. Melhor resultado global para *SJC3b.dat*

e continuam a melhorar conforme se continua a aumentar o número de formigas, porém menos intensamente. Isso pode ser explicado pelo tamanho do problema referente aos dados de *SJC1.dat*, que por possuir menos pontos e medianas já é capaz de alcançar bons resultados com menos formigas. As figuras 6 e 7 também mostram isso, uma vez que um maior número de formigas melhora o resultado, e pelo problema possuir mais pontos, 90 formigas melhorar mais o resultado em relação ao número anterior do que para a base *SJC1.dat*.

As figura 8 mostra que aumentar o número de formigas de 60 para 90 não melhorou muito a solução, inclusive fazendo-a piorar por volta da iteração 300 em diante. Porém, as figuras 9 e 10 mostram que a melhor solução local e a média das soluções, respectivamente, melhoraram ao se aumentar o número de formigas de 60 para 90. É possível então que a falta de melhoria do melhor global não seja o mais esperado.

Devido aos experimentos com 90 formigas no geral não terem apresentado uma melhora tão significativa em relação a 60 formigas, não foram feitos experimentos com mais formigas. Apesar do resultado para a base *SJC3b.dat* ter piorado levemente com 90 formigas, a piora foi tão pequena que, para facilitar os experimentos posteriores, foi definido o número de formigas para todas as bases como 90.

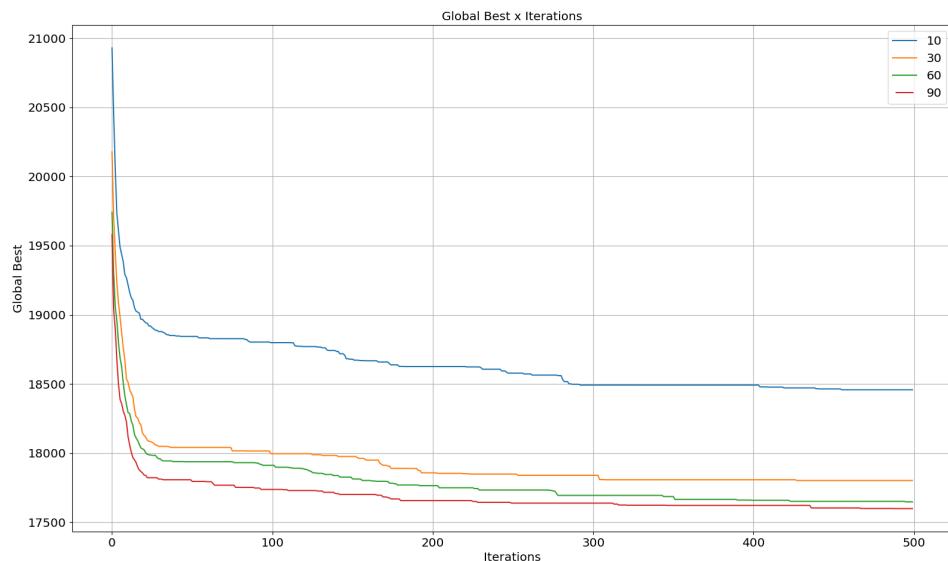


Figure 4. Melhor solução global para *SJC1.dat* variando número de formigas.

4.2.3. Variação dos valores de alfa e beta

O objetivo do terceiro teste realizado foi analisar a influência na qualidade da solução proveniente dos valores de α , influência do feromônio τ_i na função de transição, e β , influência do valor de η_i na função de transição.

Número de Iterações: 500

Número de Formigas: 90

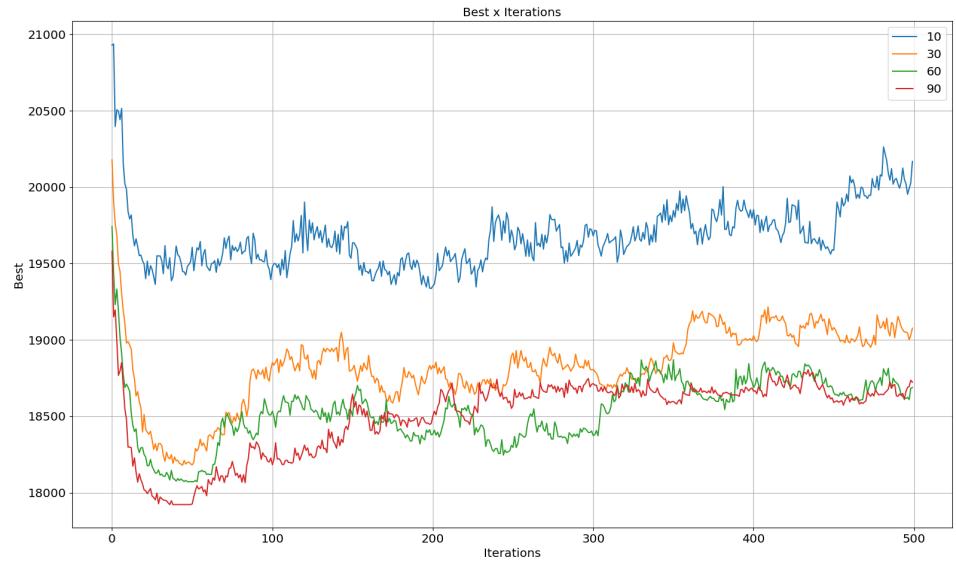


Figure 5. Melhor solução da iteração para **SJC1.dat** variando número de formigas.

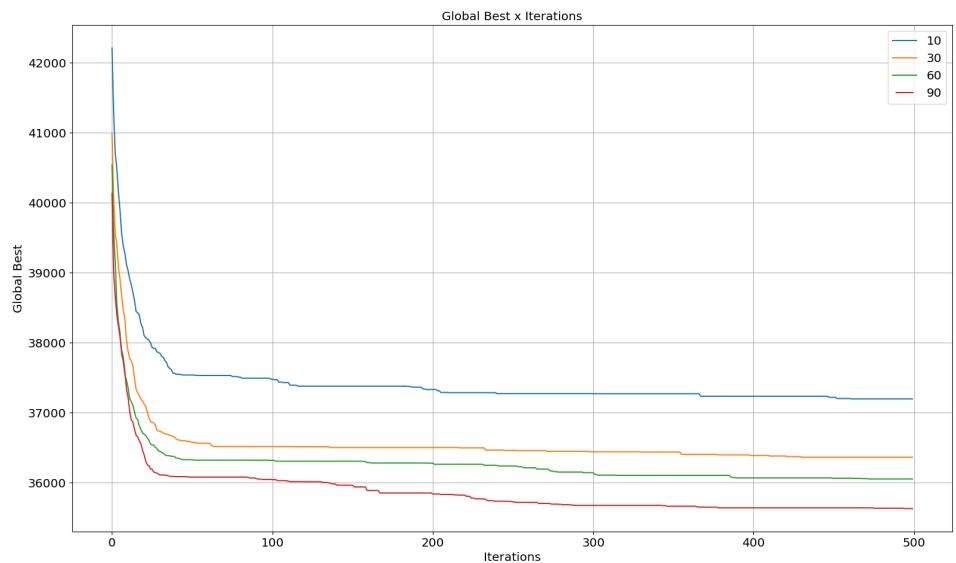


Figure 6. Melhor solução global para **SJC2.dat** variando número de formigas.

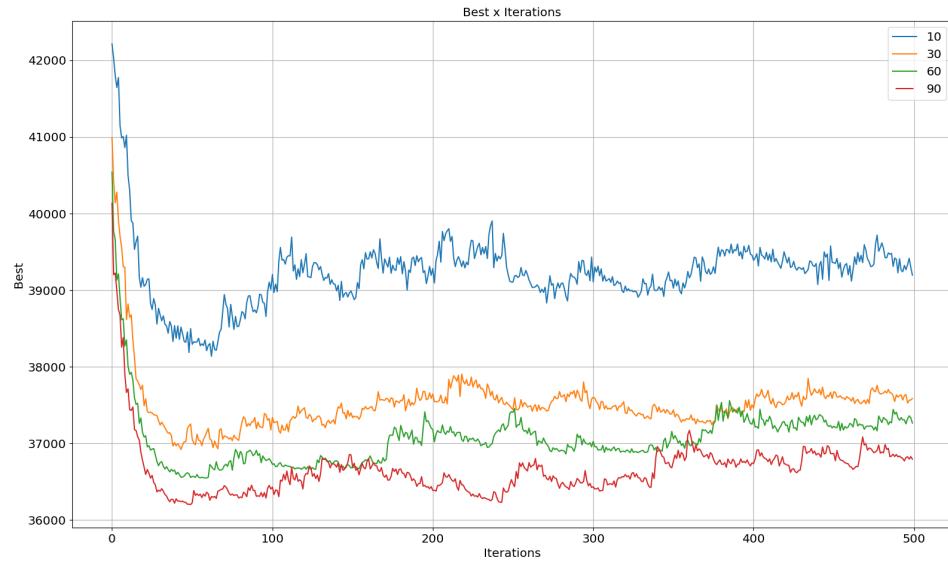


Figure 7. Melhor solução da iteração para **SJC2.dat** variando número de formigas.

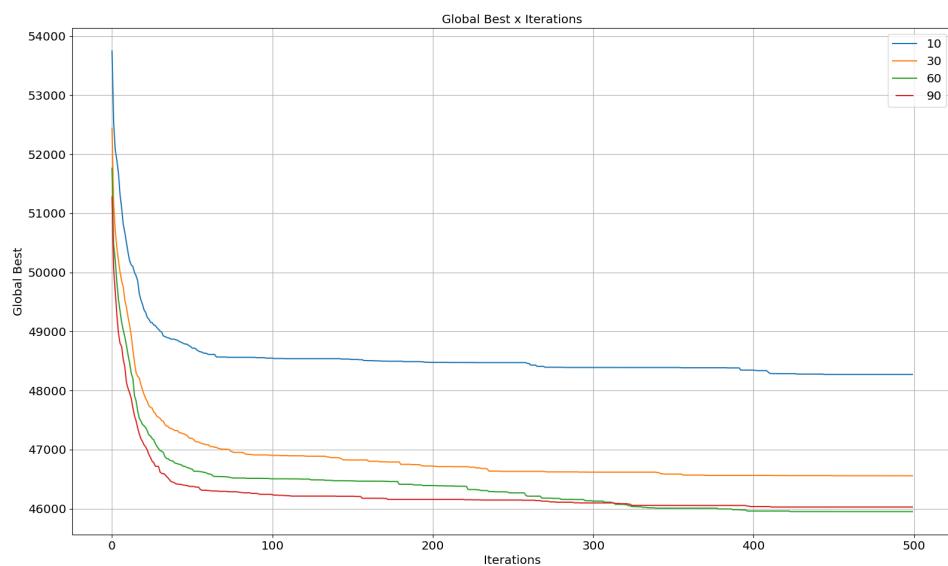


Figure 8. Melhor solução global para **SJC3b.dat** variando número de formigas.

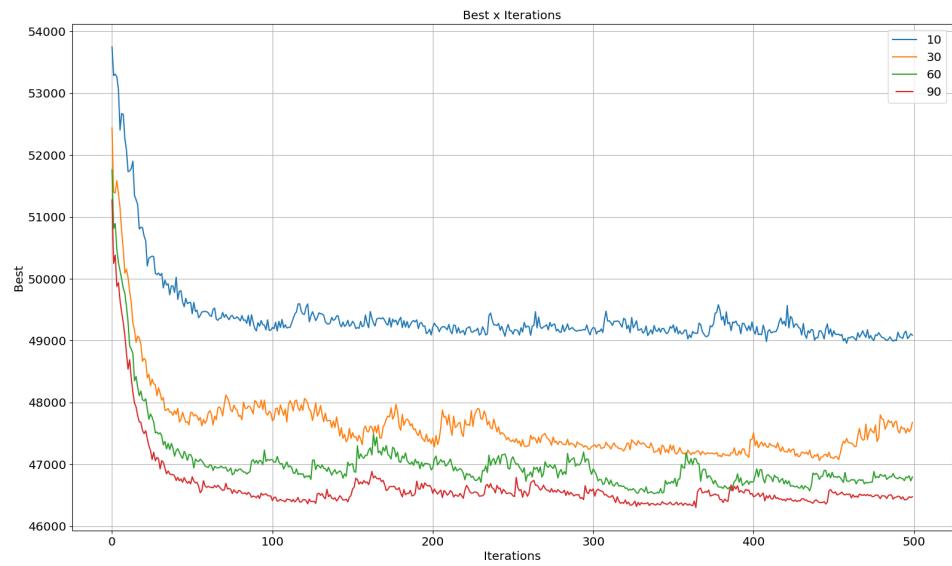


Figure 9. Melhor solução da iteração para *SJC3b.dat* variando número de formigas.

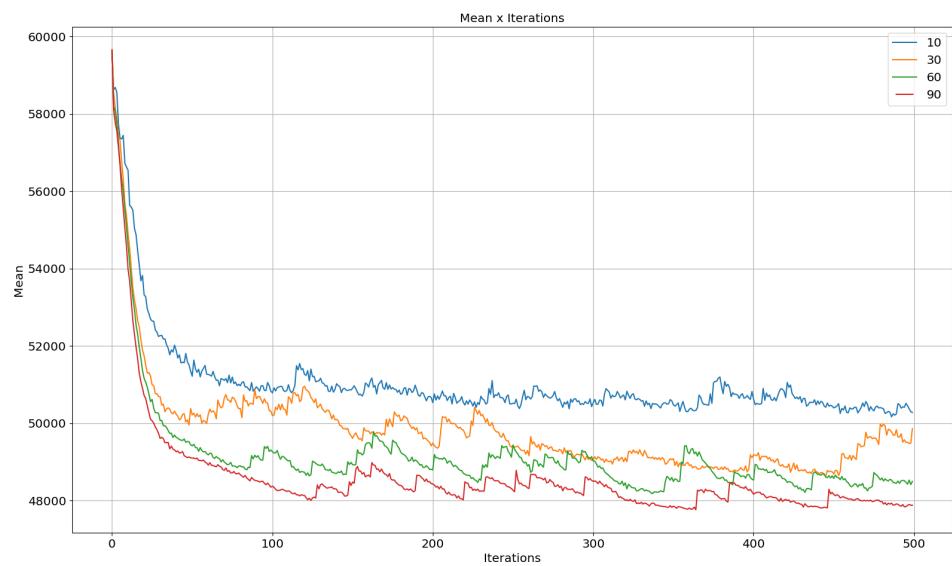


Figure 10. Média das soluções da iteração para *SJC3b.dat* variando número de formigas.

Taxa de evaporação: 0.1

Foram testadas as seguintes combinações de valores:

α	β
3	1
1	0
1	3
0	1

A função heurística η , apesar de fornecer uma boa estimativa sobre os melhores candidatos para medianas, pode não representar um cenário adequado. Dessa forma, esperava-se que dar mais ênfase ao valor de α do que de β levaria a resultados melhores. Os valores para o melhor resultado global, melhor resultado local e média dos resultados para a base *SJC1.dat* mostrados nas figuras 11, 12 e 13 respectivamente, mostram que um α maior do que o β resulta em resultados melhores. É interessante observar porém, que as melhores soluções ocorrem com $\alpha = 1, \beta = 0$, enquanto a média dos melhores resultados ocorre com $\alpha = 3, \beta = 1$. Isso sugere que a heurística da função η é eficiente em sugerir melhores resultados em média, mas a melhor solução para esse conjunto de dados não remete a essa sugestão, de forma que ao ignorar a função η obtém-se uma melhor solução global.

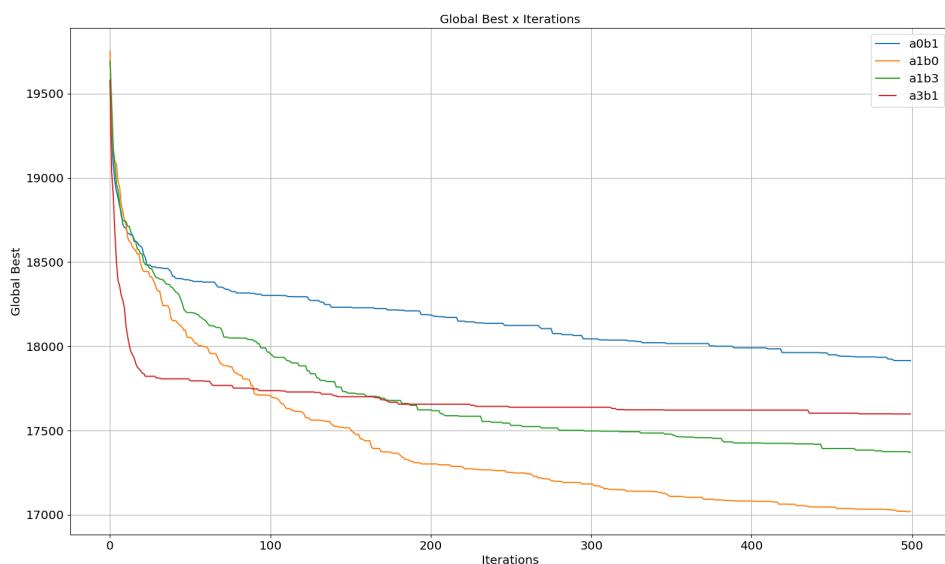


Figure 11. Melhor solução global para *SJC1.dat* variando os valores de alfa e beta.

Esse tipo de comportamento se repete para os conjuntos de dados *SJC2.dat* e *SJC3.dat*. A base *SJC2.dat* tem a melhor solução global para $\alpha = 1, \beta = 0$, como se vê na figura 14, mas a melhor solução local, mostrada na figura 15, e a média das soluções, mostrada na figura 16, apresentam melhores resultados sempre com $\alpha = 3, \beta = 1$. Isso significa que, no geral, para essa base, é possível obter uma boa média com a função η , porém não a solução ótima.

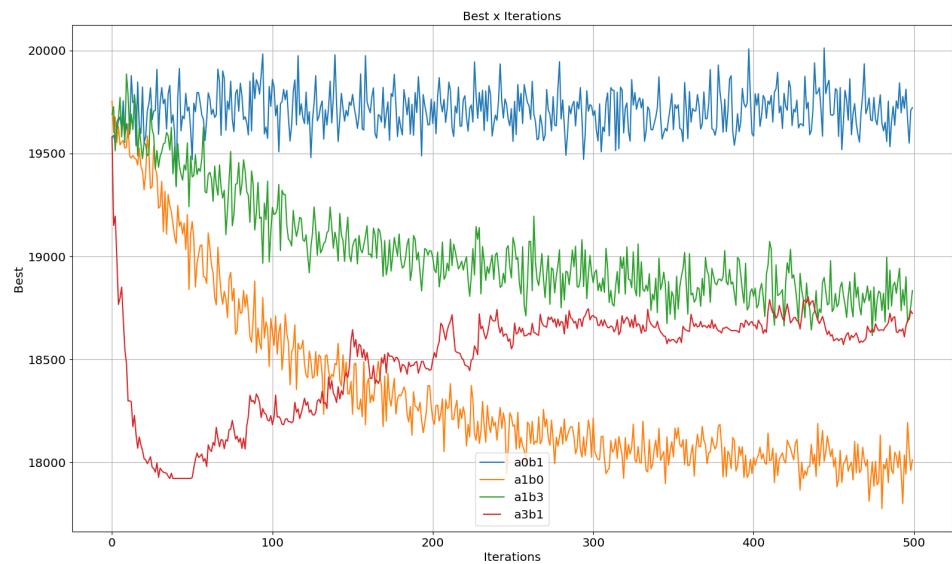


Figure 12. Melhor solução da iteração para **SJC1.dat** variando os valores de alfa e beta.

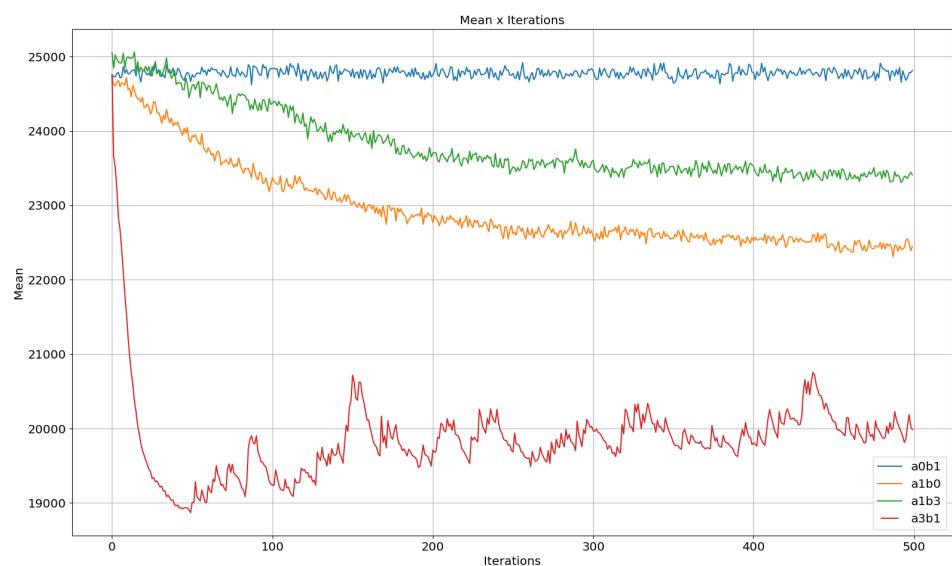


Figure 13. Média das soluções da iteração para **SJC1.dat** variando os valores de alfa e beta.

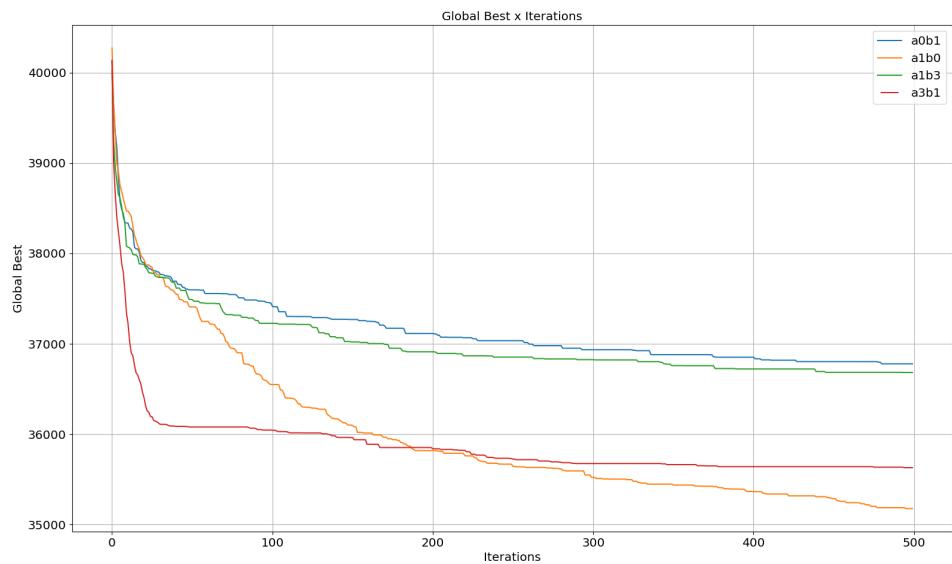


Figure 14. Melhor solução global para **SJC2.dat** variando os valores de alfa e beta.

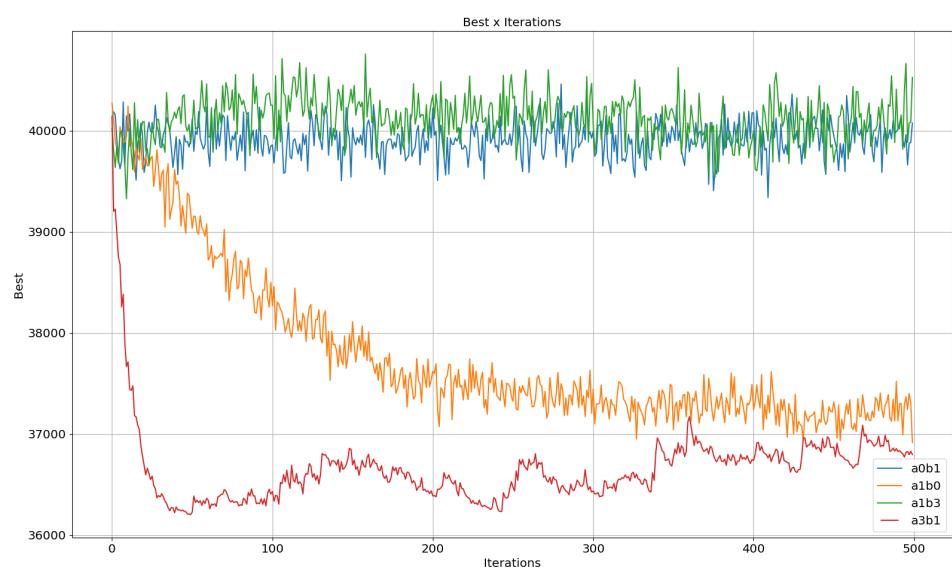


Figure 15. Melhor solução da iteração para **SJC2.dat** variando os valores de alfa e beta.

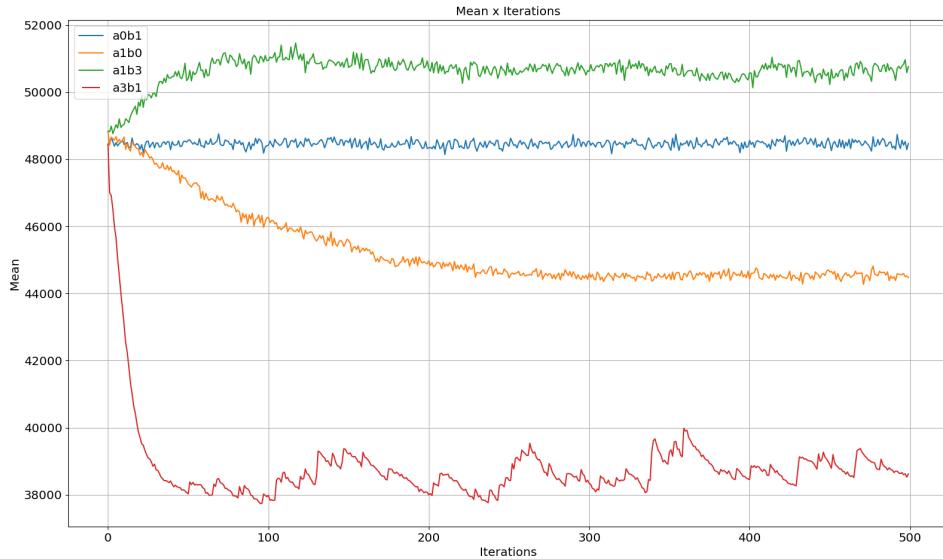


Figure 16. Média das soluções da iteração para *SJC2.dat* variando os valores de alfa e beta.

A base *SJC3b.dat*, apesar de também apresentar um melhor resultado global com os valores $\alpha = 1, \beta = 0$, apresenta uma melhor solução global muito próxima para os valores $\alpha = 3, \beta = 1$, como mostra a figura 17. Os valores da melhor solução local, na figura 9, média das soluções, na figura 19, são sempre muito melhores para $\alpha = 3, \beta = 1$. Isso leva a conclusão de que a heurística é boa o suficiente pra achar um valor tão bom quanto sem ela.

Em todas as bases de dados é possível notar que a heurística faz com que um resultado bom seja encontrado mais cedo do que sem a mesma. Nas ocasiões em que não usar a heurística apresenta um melhor resultado, esse resultado superior só é encontrado mais tarde nas iterações.

Uma vez que os valores $\alpha = 1, \beta = 0$ levaram às melhores soluções globais, eles foram definidos como os valores padrões para o próximo teste.

4.2.4. Variação da taxa de evaporação

Nesse teste, a taxa de evaporação do feromônio foi variada.

Número de Iterações: 500

Número de Formigas: 90

Alfa: 1

Beta: 0

Os valores testados para a taxa de evaporação foram 0.1, 0.2, 0.3, 0.4 e 0.5.

As figuras 20, 21 e 22 mostram, respectivamente, a melhor solução global, a melhor solução local e média das soluções para a base *SJC1.dat*. Observa-se que, conforme

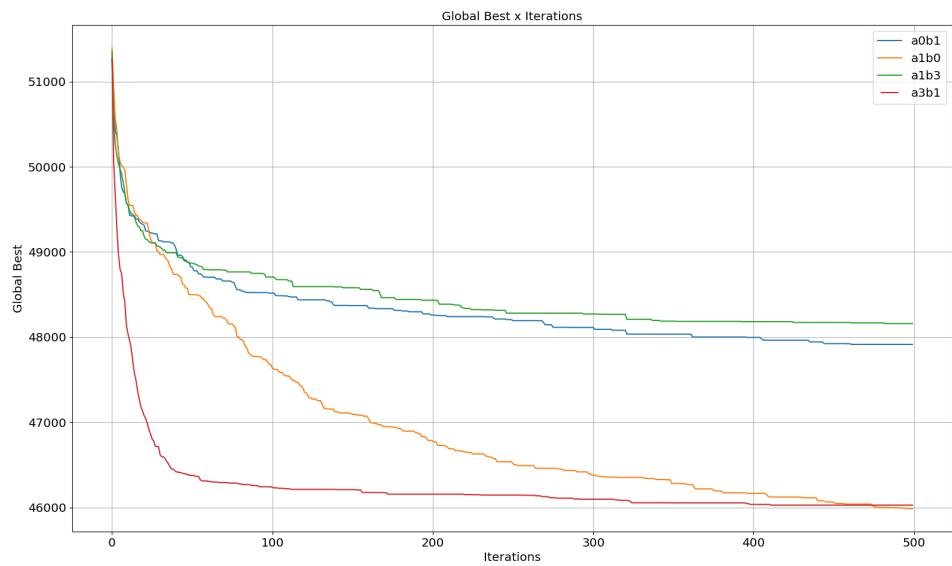


Figure 17. Melhor solução global para **SJC3b.dat** variando os valores de alfa e beta.



Figure 18. Melhor solução da iteração para **SJC3b.dat** variando os valores de alfa e beta.

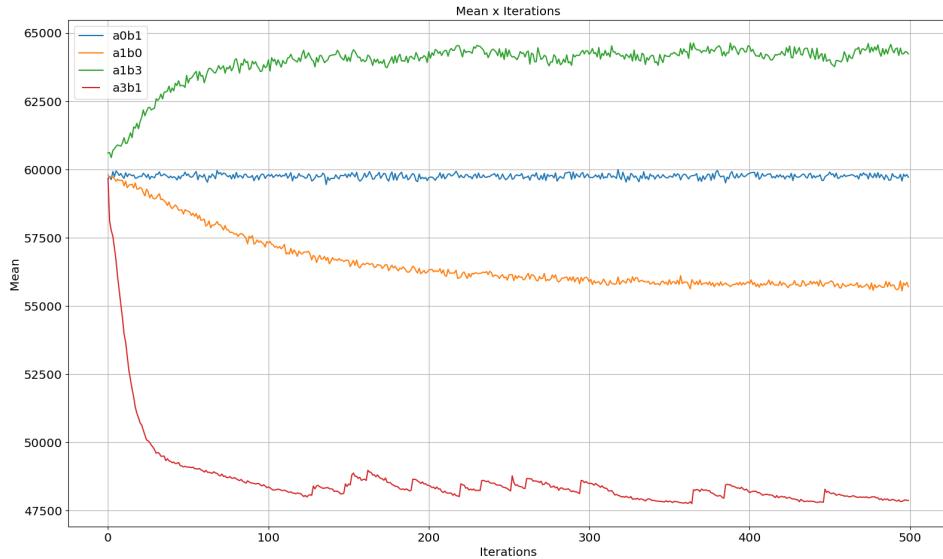


Figure 19. Média das soluções da iteração para *SJC3b.dat* variando os valores de alfa e beta.

a taxa de evaporação aumenta, a qualidade das soluções também aumentam. Isso sugere que, uma maior taxa de evaporação permite que o espaço de busca seja explorado de forma mais eficaz, encontrando melhores soluções.

As figuras 23, 24 e 25 mostram, respectivamente, a melhor solução global, a melhor solução local e média das soluções para a base *SJC2.dat* e as figuras 26, 27 e 28 mostram a melhor solução global, local e média das soluções para a base *SJC3b.dat*. Os resultados apresentados para essa base reforçam as conclusões tomadas para a base *SJC1.dat*.

Após uma análise dos resultados, o parâmetro de evaporação ótimo escolhido foi 0.5.

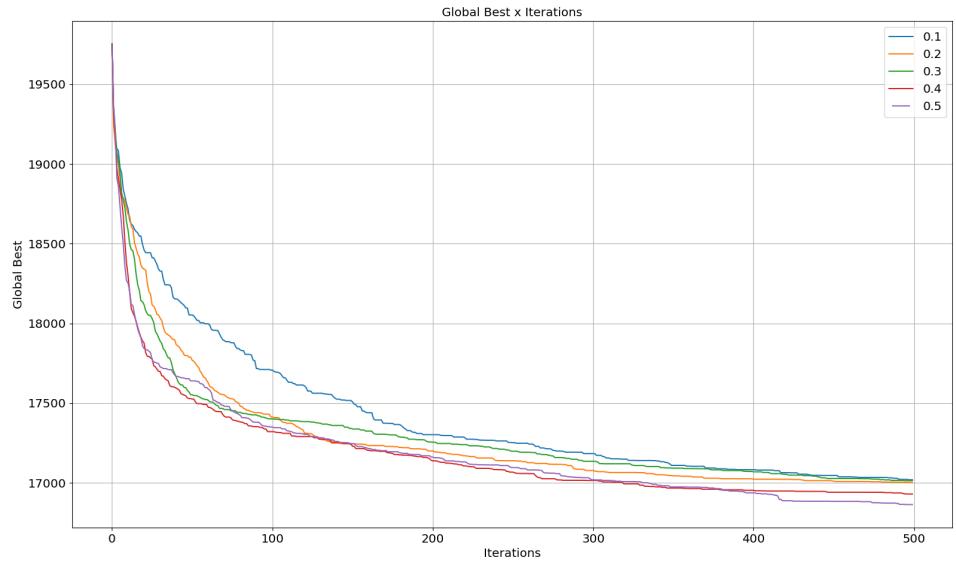


Figure 20. Melhor solução global para **SJC1.dat** variando a taxa de evaporação.

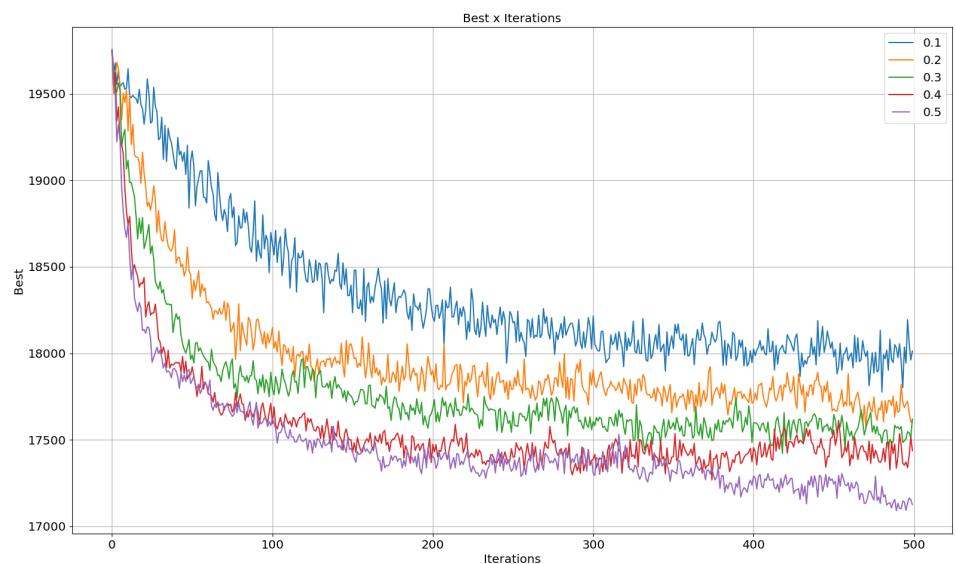


Figure 21. Melhor solução da iteração para **SJC1.dat** variando a taxa de evaporação.

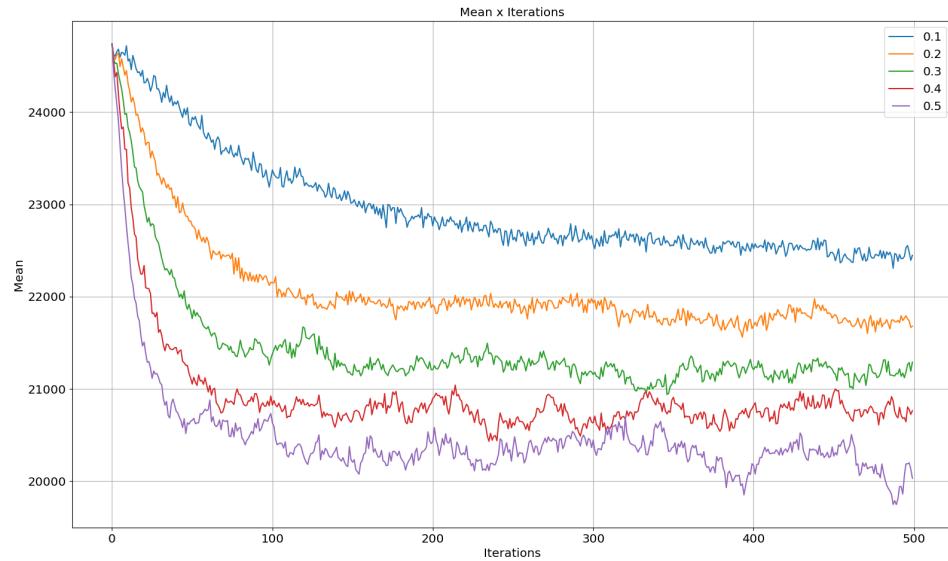


Figure 22. Média das soluções da iteração para **SJC1.dat** variando a taxa de evaporação.

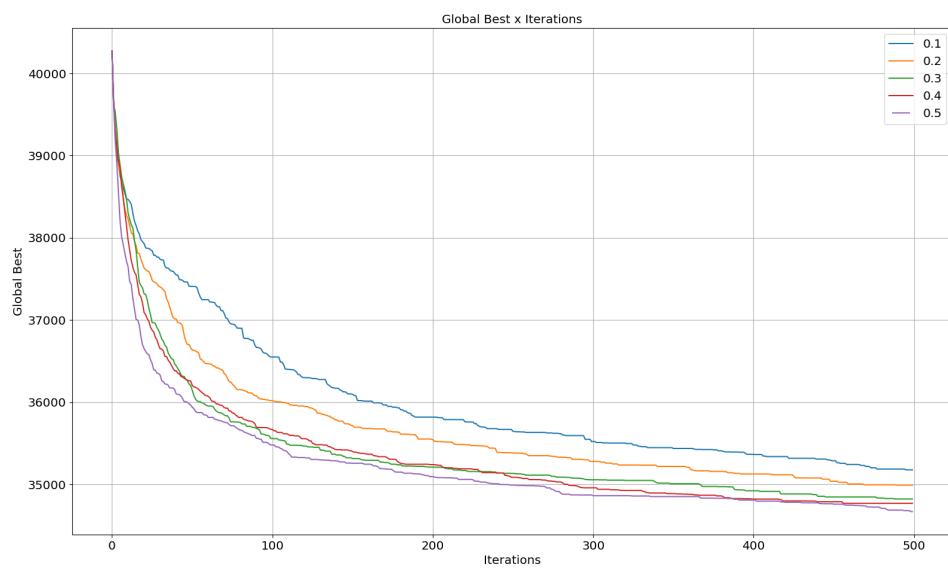


Figure 23. Melhor solução global para **SJC2.dat** variando a taxa de evaporação.

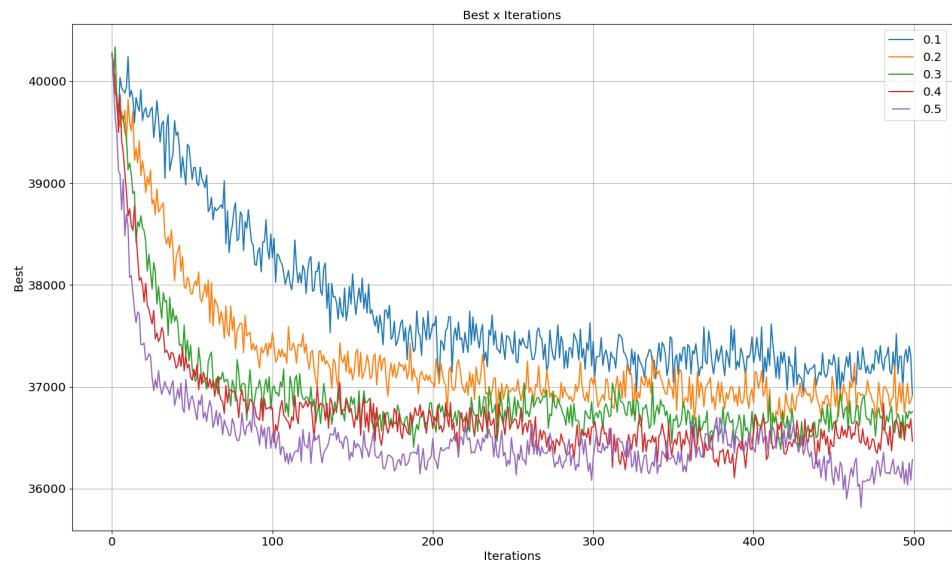


Figure 24. Melhor solução da iteração para *SJC2.dat* variando a taxa de evaporação.

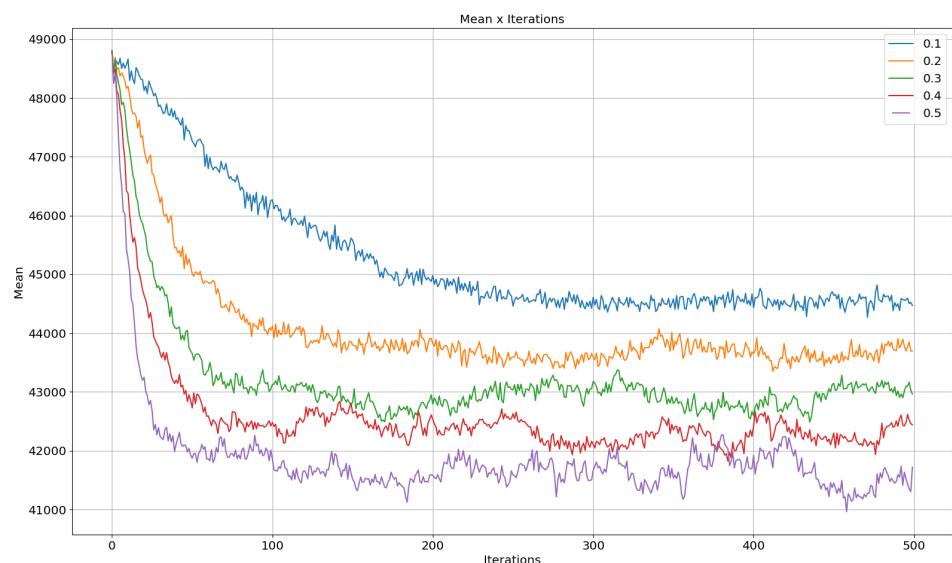


Figure 25. Média das soluções da iteração para *SJC2.dat* variando a taxa de evaporação.

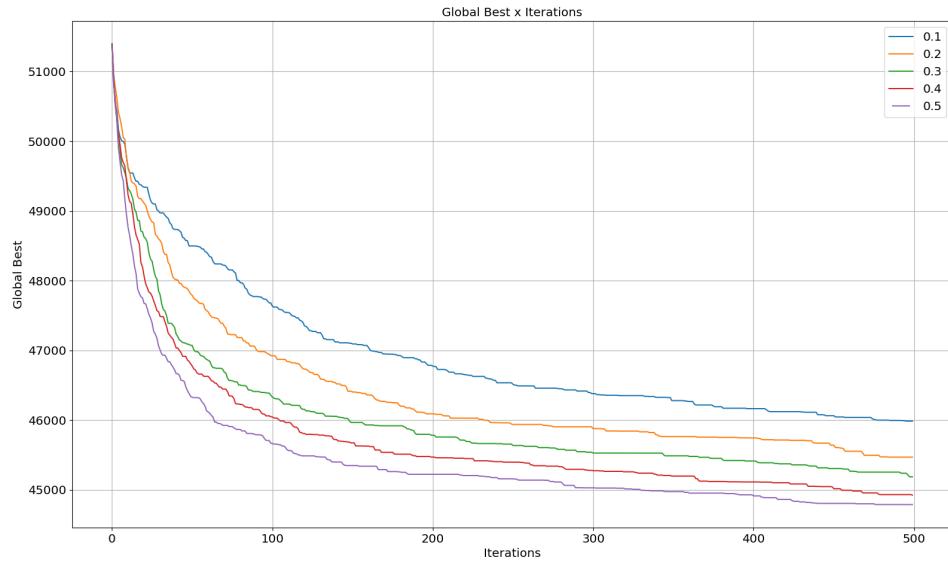


Figure 26. Melhor solução global para **SJC3b.dat** variando a taxa de evaporação.

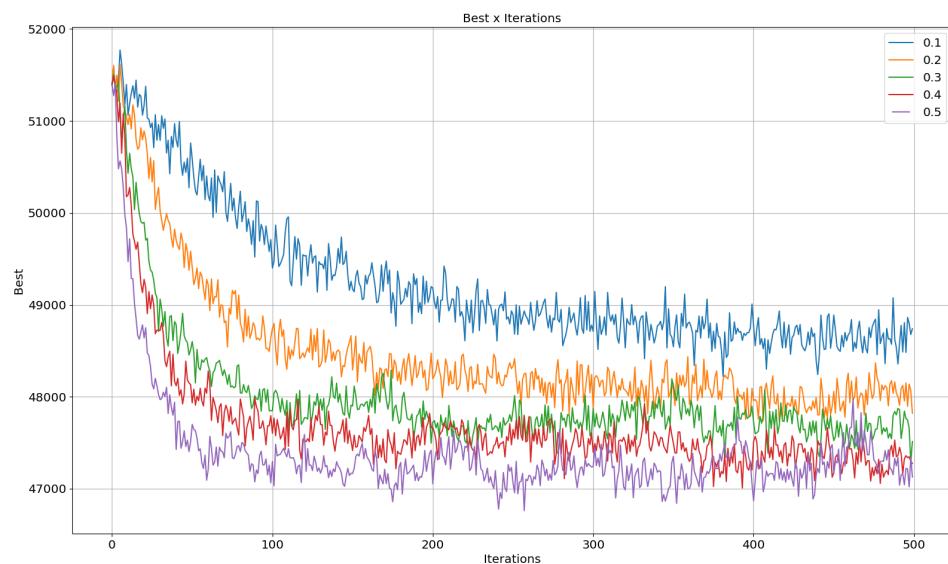


Figure 27. Melhor solução da iteração para **SJC3b.dat** variando a taxa de evaporação.

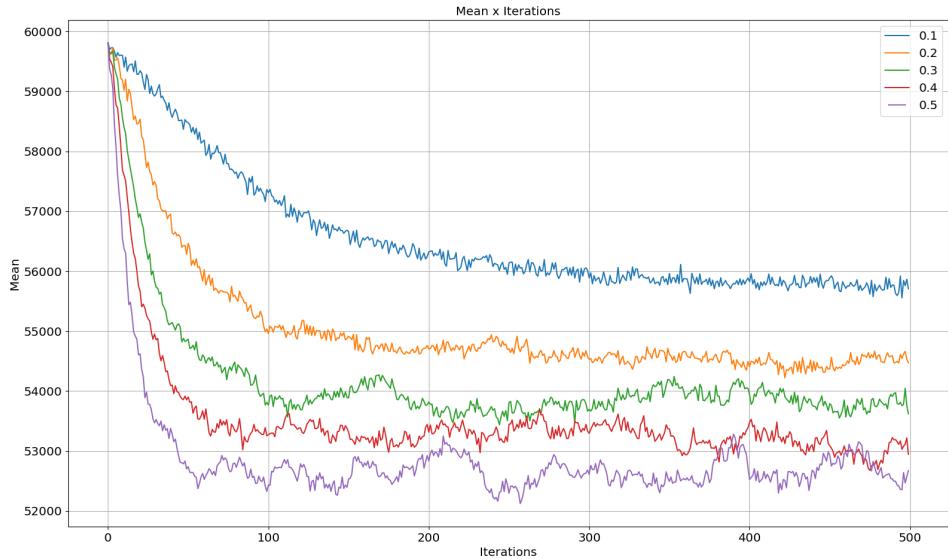


Figure 28. Média das soluções da iteração para *SJC3b.dat* variando a taxa de evaporação.

4.3. Resultados

A comparação dos resultados, obtidos ao utilizar os melhores parâmetros para a base de dados *SJC1.dat* ao longo das iterações, com o valor ótimo fornecido pode ser observada na figura 29. O melhor valor obtido pela colônia de formigas foi显著mente melhor do que o ótimo fornecido. Uma possível explicação seria a possibilidade da heurística utilizada para alocar os clientes para uma mediana não ser capaz de alocar todos os clientes, e assim a soma total das distâncias ser menor por não incluir todos os clientes. Outra possibilidade seria um erro no valor fornecido como ótimo, o que, nesse caso, significaria que o algoritmo foi capaz de encontrar um valor mais próximo do ótimo real, ou até mesmo o próprio ótimo.

A comparação do ótimo da base *SJC2.dat*, mostrada na figura 30, evidencia que o algoritmo não foi capaz de atingir o ótimo fornecido, porém a diferença entre as somas da distância foi de menos de 2000. Levando em consideração que o valor fornecido corresponde a um *lower bound*, pode-se considerar que o resultado obtido foi bom, porém, pela curva do gráfico, o algoritmo não estava próximo de chegar ao ótimo.

A comparação do ótimo da base *SJC3b.dat*, mostrada na figura 31, por sua vez, foi a que mais se distanciou do ótimo fornecido. A distância entre o ótimo e o resultado obtido não é superior a 6000, mas não caracterizou um bom resultado. Porém, como o algoritmo de colônia de formigas não funciona sozinho, e sim em conjunto com uma heurística para alocar os clientes as medianas, e não foram implementados heurísticas usadas em [de França et al. 2005] para melhorar o resultado, não se pode dizer que o algoritmo não foi eficaz em encontrar as medianas.

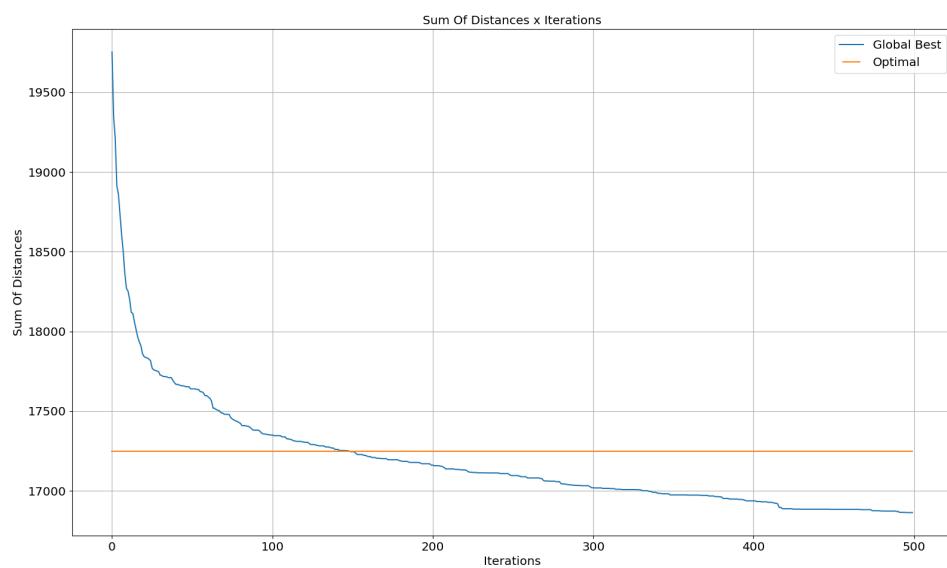


Figure 29. Comparação dos resultados obtidos para o melhor global com o ótimo fornecido para a base *SJC1.dat*.

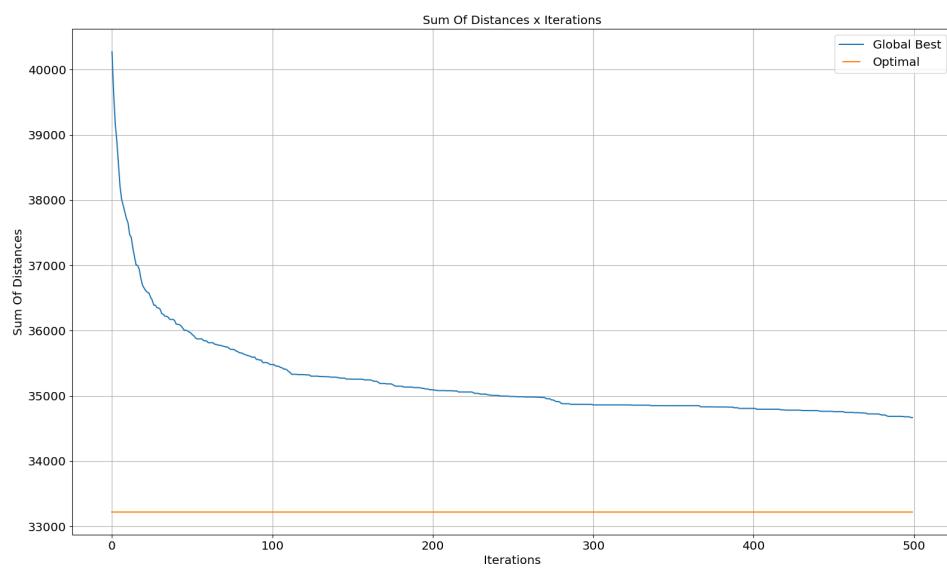


Figure 30. Comparação dos resultados obtidos para o melhor global com o ótimo fornecido para a base *SJC2.dat*.

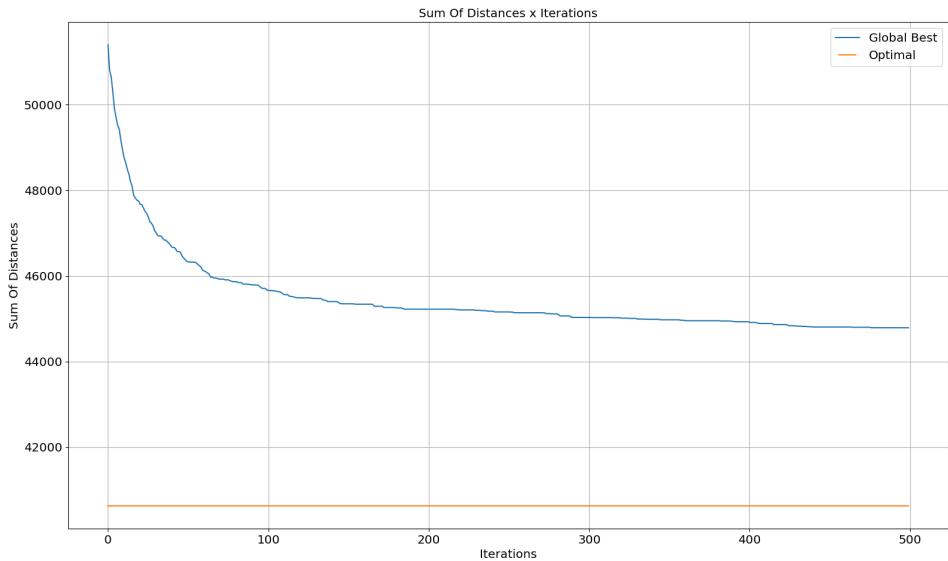


Figure 31. Comparação dos resultados obtidos para o melhor global com o ótimo fornecido para a base *SJC3b.dat*.

5. Conclusão

A tarefa de modelagem do problema apresentado mostrou que nem sempre um problema poderá ser modelado facilmente para uso em um algoritmo. O problema das p-medianas foi modelado só em parte para o algoritmo de colônia de formigas, deixando uma parcela do problema para ser resolvida por um heurística. Isso ajuda a perceber que, em certas ocasiões é necessário criatividade para se poder resolver e modelar um problema.

As escolhas dos parâmetros dos algoritmos mostrou nem sempre a escolha mais óbvia traz os melhores resultados, com destaque para os valores de alfa e beta, que levaram a encontrar a melhor solução utilizando parâmetros diferentes do que os utilizados para melhorar a solução média. O número de iterações foi importante para obter uma melhor solução, mas esse número não influenciou tanto quanto a quantidade de formigas. Apesar disso, existe um limite para o qual se pode aumentar ambos os números de forma a melhorar o resultado.

Os resultados obtidos com o algoritmo foram bons em geral, obtendo os melhores resultados para a base *SJC1.dat*. A base *SJC3b.dat* foi a que teve piores resultados, que ainda assim não tiveram uma diferença de mais de 6000 para o ótimo. Dessa forma pode-se observar que o algoritmo da colônia de formiga foi eficaz em encontrar bons resultados, apesar de não ter encontrado o resultado ótimo.

6. Bibliografia

References

- de França, F. O., Zuben, F. J. V., and de Castro, L. N. (2005). Max min ant system and capacitated p-medians: Extensions and improved solutions. *Informatica (Slovenia)*, 29(2):163–172.