A Preprocessing Phase for the Evolutionary Clustering Search

Antonio Augusto Chaves
Instituto Nacional de Pesquisas Espaciais
Av. dos Astronautas, 1758 São Paulo, Brasil
chaves@lac.inpe.br

Luiz Antonio Nogueira Lorena
Instituto Nacional de Pesquisas Espaciais
Av. dos Astronautas, 1758 São Paulo, Brasil
lorena@lac.inpe.br

Abstract

This paper approaches a recent hybrid evolutionary algorithm, called Evolutionary Clustering Search (ECS), that proposes a way of detecting promising areas combining an evolutionary algorithm and a iterative clustering. The search strategy become more aggressive in such detected areas by applying local search. In this paper, we developed a preprocessing phase for the ECS applying a location-allocation algorithm. The proposed approach is validated solving the Prize Collecting Travelling Salesman Problem.

1. Introduction

A challenge in combinatory optimization is to define efficient strategies to cover the all search space, making possible the application of local search only in really promising search areas. This paper presents a preprocessing approach for the method Evolutionary Clustering Search (ECS) [17], that consists a way of detecting promising search areas based on clustering. In the ECS, there is a genetic algorithm (GA) that generates individuals to be clustered. These promising areas should be explored through local search methods as soon as they are discovered.

The approach proposed in this paper applies a location-allocation algorithm in the initial phase of ECS, seeking to discover the clusters that represent the initial population generated by the genetic algorithm. The objective of this approach is to improve ECS, to have more representative clusters of the search space and consequently applying heuristics of local search in clusters with more efficiency.

To validate the new approach we solved the Prize Collecting Travelling Salesman Problem (PCTSP). The PCTSP is a generalization of the Travelling Salesman Problem, where a salesman collects a prize p_i in each city visited and pays a penalty γ_i for each city not visited, considering travel costs c_{ij} between the cities. The problem intend to minimize the sum of travel costs and penalties paid, while including in the tour an enough number of cities that allow collecting a minimum prize (p_{min}), defined a priori.

The remainder of the paper is organized as follows. Section 2 reviews previous works about PCTSP. Section 3 describes the method ECS and the modifications. Section 4 present the location-allocation algorithm and the section 5 present ECS applied to PCTSP. Section 6 presents the computational results and section 7 concludes the paper.

2. Literature review

The PCTSP was introduced by Balas [2] as a model for scheduling the daily operations of a steel rolling mill. The context of steel rolling mill can be described in the following way. A rolling mill produces steel sheets from slabs by hot or cold rolling and schedulers have to choose from an inventory a collection of slabs and order it so as to minimize some function of the sequence. The author presented some structural properties of the problem and two mathematical formulations.

Fischetti and Toth [9] developed several bounding procedures, based on different relaxations (e.g. Lagrangean relaxation, disjunction and instance transformation). A branch and bound algorithm was also developed which was applied to small size problems.

Goemans and Williamson [11] provide a 2-aproximation procedure to a version of the PCTSP, in which the minimum prize to be collected is removed and the objective is simply to minimize the sum of travel costs and penalties paid.

Dell'Amico, Maffioli and Sciomanchen [6] developed a Lagrangean heuristic, which use a Lagrangean relaxation for generating starting solutions for the heuristic procedure. A procedure called Adding-Nodes was used to obtain feasible solutions for PCTSP through the lower bound and a procedure called "Extension and Collapse" seeks improving feasible solutions.

Gomes, Diniz and Martinhon [12] and Melo and Martinhon [15] present hybrid metaheuristics to solve the PCTSP. The first combines Greedy Randomized Adaptive Search Procedure (GRASP) [8] and Variable Neighborhood Descent (VND) [16] and the second combines GRASP

and Variable Neighborhood Search (VNS) [16] as a local search. Torres and Brito [21] present a new mathematical formulation to PCTSP based on the formulation presented in [2]. In this formulation a new group of constraints is proposed to prevent sub-tours.

Chaves et al. [3] explored two approaches. A mathematical formulation to PCTSP solved for small instances, through the solver Lingo [19], and a heuristic procedure, combining the metaheuristics GRASP and VNS. Chaves and Lorena [4] proposed new heuristics to solve the PCTSP, using the ECS and an adaptation of this, called *CS, where the evolutionary component is substituted by the metaheuristics GRASP and VNS.

Feillet, Dejax and Gendreau [7] present a survey on TSP with profits that include the PCTSP, identifying and comparing different classes of applications, modelling approaches and exact or heuristic solution techniques.

3. Evolutionary Clustering Search

The Evolutionary Clustering Search (ECS) is an evolutionary technique proposed by Oliveira and Lorena [17] that employs clustering for detecting promising areas of the search space. It is particularly interesting to find out such areas as soon as possible to change the search strategy over them. In the ECS, a clustering process is executed simultaneously to an evolutionary algorithm, identifying groups of individuals that deserve special interest.

The ECS attempts to locate promising search areas by framing them by clusters. A cluster is defined by the tuple $\mathcal{G} = \{c; r; s\}$ where c, r and s are, respectively, the center and the radius of the area, and a search strategy to be associated to the clusters.

The center is an individual that represents the cluster, identifying the location of the cluster inside of the search space. The radius establishes the maximum distance, starting from the center, that an individual can be associated to the cluster. The search strategy is a systematic search intensification, in which individuals of a cluster interact among themselves along the clustering process generating new individuals.

The ECS consists of four conceptually independent components with different attributions:

- an evolutionary algorithm (EA);
- an iterative clustering (IC);
- an analyzer module (AM);
- a local searcher (LS);

The EA works as a full-time solution generator. The population evolves independently of the remaining components. Individuals are selected, crossed over, and updated for the next generations. Simultaneously, clusters are maintained to represent these individuals.

The IC is the kernel of ECS, working as a classifier of information (solutions represented by individuals) into groups, maintaining in the system just information that are relevant for the process of search intensification. IC is designed as an iterative process that forms groups by reading the individuals being selected or updated by EA.

The AM provides an analysis of each cluster, in regular intervals of generations, indicating a probable promising cluster. Typically, the density of the cluster is used in this analysis, that is, the number of selections or updating that happened recently in the cluster. A cluster with high density should have a promising center. AM is also responsible for the elimination of clusters with lower densities.

Finally, the LS is a local search module that provides the exploration of a supposed promising search area. This process happens after the component AM has discovered a promising cluster. The local search is applied on the center of the cluster.

This paper proposes a preprocessing phase applying a location-allocation algorithm in the initial population of the genetic algorithm, seeking to find initial clusters to represent the search space of the problem appropriately. In the original ECS approach, the clusters were only discovered in the evolution process of the population.

Figure 1 shows the four components of the ECS, the location-allocation phase, the population and the clusters of individuals.

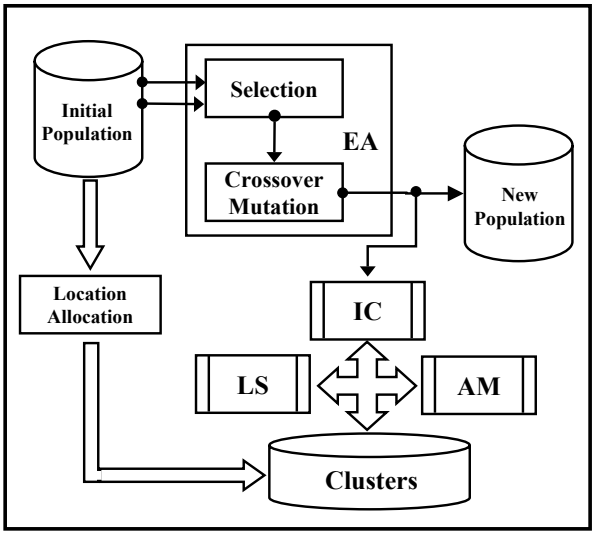

Figure 1. ECS components

The initial population of the EA usually meets very dispersed in the search space, being randomly generated. Therefore, the initial clusters can represent the search space

appropriately, improving the search process for promising clusters.

4. Location-Allocation Algorithm

The location-allocation algorithm proposed in this paper is based in an algorithm used in [14] for the maximal covering location problem, that consists in searching for a new center in each cluster, swapping the current center with a non-center of the same cluster, changing the allocation solution.

The inputs of the algorithm are the individuals of the initial population generated by the genetic algorithm, and the number $\mathcal{MC}$ of clusters. The maximum number of clusters was defined a priori, through empiric tests, as being 20 clusters.

The metric distance used in this algorithm is the number of different edges between the individual and the center of the cluster, and, as larger the number of different edges among themselves, more the dissimilarity.

The first step of the location-allocation algorithm is to randomly generate the clusters' centers and allocate each individual of the initial population to the cluster with nearest center. After this, the average of the distances is calculated from the individuals to the centers of their clusters. This average is the function used to qualify de clusters'centers.

The following step is to determine new centers for the clusters, swapping the current center with a individual of the same cluster, changing the allocation solution. The individuals of the population are reallocated to the clusters and the new average of the distances are calculated. This step is repeated by a certain number of iterations and at the end the centers that provide the best allocation are returned.

The interchange procedure for centers in each cluster are performed only for individuals inside of a radius of covering of the initial center.

Figure 2 presents a pseudo-code for the location-allocation algorithm.

```
procedure Location-Allocation
    randomly initialize clusters centers
    allocate all individuals to the nearest center
    calculate the average of the distances
    while (number of iterations not satisfied) do
        interchange center and individual in cluster C^k
        reallocate all individuals in the new centers
        recalculate the average of the distances
    end while
    return the best centers
end procedure
```

Figure 2. Location-Allocation code

Figure 3 presents a example of initial clusters and a iteration of the location-allocation algorithm.

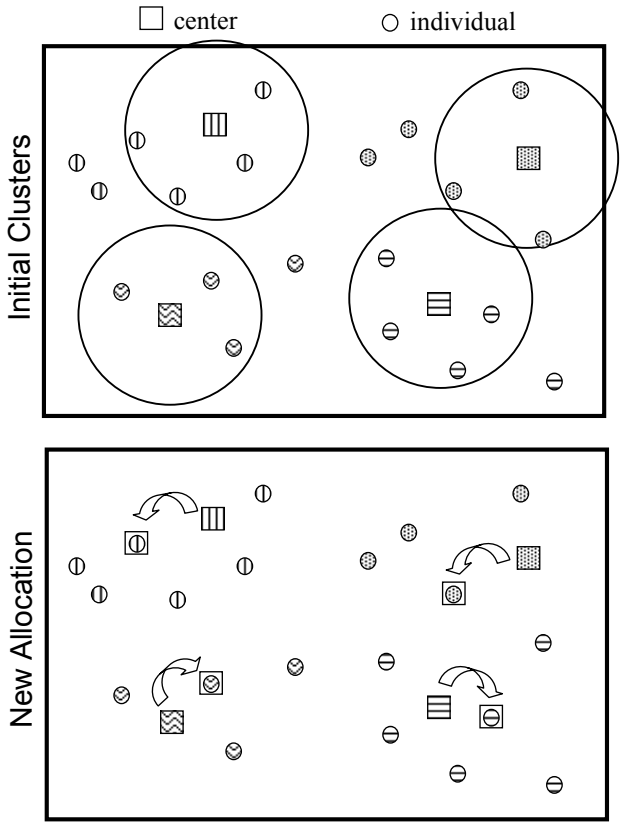

Figure 3. Example of Location-Allocation

5. ECS for PCTSP

A revised version of the ECS for the Prize Collect Travelling Salesman Problem (PCTSP) [4] is presented in this section. The application details are now described, clarifying the approach.

An individual was represented through a vector that contains the nodes of the problem in the order that they are visited. Observe that negative signs indicate not visited nodes. The individual representation is shown in Figure 4, where the sequence of visits is $\{1, 3, 0, 4\}$ and the nodes 5 and 2 were not visited.

1	3	-5	0	4	-2

Figure 4. Individual representation

The component EA, responsible for generating solutions to clustering process, was the Population Training Algorithm (PTA) [18] employing well-known genetic operators

as the base-guide selection [13], the crossover BOX [20] and the mutation 2-Opt.

The PTA works with a dynamic population of individuals, and, initially the population is randomly generated. All individuals are evaluated by two functions, f and g. The first evaluates the quality of individual and the second applies a problem-specific heuristic (called training heuristic) to evaluate the neighborhood of individual, being the value of the best solution found attributed to g.

In this paper, the training heuristic used to determine the desired characteristics in training along the evolutionary process of PTA is the method SeqDrop-SeqAdd [12], that consists of applying a sequence of node removal while the objective function value is being decreased and a sequence of node additions while some improvement is attained.

In each generation a constant number of individuals ($\mathcal{NS}$) is selected. Two individuals are selected for recombination and the selection is accomplished privileging the individuals with greater quality. The first individual is called the base and it is randomly selected from the best individuals of the population. The second individual is called the guide and is randomly selected out of the entire population. They are recombined by a variant of the Order Crossover (OX) called Block Order Crossover (BOX). The base and guide are mixed into one offspring, copying random blocks of both parents. Pieces copied from a parent are not copied from other, keeping the offspring feasible. Eventually, the offspring can suffer mutation.

The adaptation of an individual is proportional to its ranking δ,

$$\delta = d.\left[G_{\max} - g\right] - |f - g| \tag{1}$$

that is composed by:

- a component concerning the adaptation of individual in relation to the training heuristic: minimizing $(f - g)$;
- a component that privileges the minimization of the function g, thought the minimization of a distance between the individual and an estimate of an upper bound for all the possible values that the functions f and g can assume: the constant G_{max};
- and a constant d, $0 \leq d \leq 1$, to balance the two components of equation 1;

So, better individuals have greater ranks.

The population is then controlled in a dynamic way by an adaptive rejection threshold, τ,

$$\tau_{i+1} = \tau_i + \xi.\left|P\right|.\frac{\left(\delta_1 - \delta_{|P|}\right)}{RG} \tag{2}$$

that is updated during the evolutionary process. Expression 2 uses the current population size, $|P|$, the best (δ_1) and worst $(\delta_{|P|})$ rankings of individuals in current population, the estimated remaining number of generations, $\mathcal{RG}$, and the ξ constant that controls the speed of the evolutionary process. At the end of each generation the individuals less adapted $(\delta \leq \tau)$ are eliminated from the population.

The component IC executes an iterative clustering of each offspring generate by the PTA. Whenever a offspring is far away from all centers, a new cluster must be created. Otherwise, the offspring should cause a disturbance (assimilation) in the most similar center. Again, the metric distance is the number of different edges among the offspring and the center of the cluster.

The assimilation process uses the path-relinking method [10], which accomplishes exploratory movements in the path that connects the individual and the center of the cluster. Therefore, the assimilation process is already a method of local search inside of the cluster.

The component AM is executed whenever an individual is assigned to a cluster, verifying if the cluster can be considered promising. A cluster becomes promising when reaches a certain *density* λ,

$$\lambda_t \geq \mathcal{PD}.\frac{\mathcal{NS}}{|C_t|} \tag{3}$$

where, $\mathcal{PD}$ is the desirable cluster population beyond the normal population, obtained if $\mathcal{NS}$ was equally divided to all clusters, and $|C_t|$ is the current number of clusters. The center of a promising cluster is refined through the component LS.

The component AM also has as function of cooling all clusters that were activated in each generation, decreasing the density of the clusters. It is also used to eliminate clusters with low density.

The component LS was implemented by the 2-Opt heuristic [5], which seeks to improve the center of a promising cluster. The 2-Opt is based on 2-changes over a complete initial solution. In this problem, a 2-change of a permutation consists of deleting 2 arcs and replacing them by 2 other arcs to form a new permutation.

Figure 5 presents the ECS pseudo-code.

6. Computational Results

The ECS was coded in C++ and was run on *Pentium 4 of 3.00 GHz* with *512 of DDR Memory*. The experiments were accomplished with objective of evidencing the improvement of the results for the ECS with location-allocation algorithm, and validate the proposed approach.

There are no available instances for PCTSP in literature. Consequently a set of problems, with $n \in \{11, 21, 31, 51, 101, 251, 501\}$, were randomly generated in the following intervals: travel cost between the nodes: $c_{ij} \in [50, 1000]$; prize associated to each node: $p_i \in [1, 100]$;

```
procedure ECS
    randomly initialize Population (P)
    Location-Allocation Algorithm
    while (number of generations not satisfied) do
        while (number of crossover) do
            selection BaseGuide (s_base, s_guide)
                s_new = crossover BOX (s_base, s_guide)
                if (mutation condition) then
                    mutation 2-Opt (s_new)
                compute f (s_new), g (s_new), δ( s_new)
                update (s_new) in P
                component IC (s_new)
                component AM (active clusters)
                if (active cluster is promising) then
                    component LS (center of the cluster)
        end while
        τ = Adaptive Increment (τ)
        for (all s_k ∈ P and δ(s_k) < τ) do
        delete (s_k) from P
    end while
end procedure
```

Figure 5. ECS code

penalty associated to each node: $\gamma_i \in [1, 750]$. The minimum prize, p_{min}, to be collected represents 75% of the sum of the prizes of all nodes. These test problems are available in http://www.lac.inpe.br/∼lorena/instancias.html.

The following parameters' values for approach ECS was adjusted through several executions. The following parameters obtained the best results.

- number of individuals selected at each generation $\mathcal{NS} = 200$;
- maximum number of clusters $\mathcal{MC} = 20$;
- population pressure $\mathcal{PD} = 2.5$;
- upper bound G_{max} is the worst value of an individual in the initial PTA population;
- increment of the rejection threshold $\xi = 0.001$;

A mathematical formulation is presented in [4] and solved using the software CPLEX 7.5 [1]. The CPLEX solved the PCTSP up to 31 nodes in a reasonable execution time. However, for the larger problems, the CPLEX took several days execution to find the optimal solution. A problem with 101 nodes (*v100a*) was executed, using an upper bound, by more than three days and did not get to close the gap between lower and upper bounds.

Table 1 presents the computational results found by the ECS with location-allocation algorithm (ECS$_{loc}$) and just for ECS applied to PCTSP. This table also show the results of the CPLEX found in [4]. The best solutions found (BS) and the execution time in seconds (ET) were considered to compare the approach performances. The values in bold indicate which approach have better objective function values and execution times for each problem.

Table 1. Results of the experiments

		CPLEX			ECS		ECS$_{loc}$	
Problem	$\|V\|$	**BS**	**ET(s)**	***gap***	**BS**	**ET(s)**	**BS**	**ET(s)**
v10	11	1765	0.06	0	**1765**	**0.03**	1765	0.62
v20	21	2302	3.73	0	**2302**	**0.72**	2302	2.87
v30a	31	3582	34.06	0	3647	208.28	**3582**	**93.64**
v30b	31	2515	45.59	0	2639	256.22	**2515**	**199.95**
v30c	31	3236	164.58	0	3236	181.17	**3236**	**146.37**
v50a	51	4328	433439.97	0	4399	464.59	**4328**	**395.51**
v50b	51	3872	241307.43	0	3942	423.67	**3872**	**310.17**
v100a	101	6879	153059.09	2.46	7395	1167.84	**6920**	**915.39**
v250a	251	-	-	-	16200	2732.97	**15450**	**2548.84**
v500a	501	-	-	-	29752	5532.68	**28790**	**3693.31**

According to Table 1, the ECS without location-allocation algorithm is faster for the small instances, that are easy to solve. However for the larger instances this approach does not find the best solutions. The ECS with location-allocation algorithm finds the optimal solution for instances up to 51 nodes in small execution times, founding better solutions for the others instances, but we can not say how close are of the optimal solution.

For each test problem, the ECS$_{loc}$ was run 10 times with different seeds and the seed that found the best solution was used in ECS execution. Table 2 shows the improvement obtained by ECS with location-allocation algorithm related to the ECS.

Table 2. Comparison among the approaches

Problem	**ECS**	**ECS$_{loc}$**	**% improvement**
v10	1765	1765	0.00
v20	2302	2302	0.00
v30a	3647	3582	1.78
v30b	2639	2515	4.70
v30c	3236	3236	0.00
v50a	4399	4328	1.61
v50b	3942	3872	1.78
v100a	7395	6920	6.42
v250a	16200	15450	4.63
v500a	29752	28790	3.23

7. Conclusions

This paper proposed a preprocessing phase for the ECS method, applying a location-allocation algorithm in the initial population of the embedded genetic algorithm. The

ECS uses the concept of hybrid algorithms, combining metaheuristics with a clustering process, detecting promising search areas. Whenever an area is considered promising some aggressive search strategy is accomplished in this area.

The ECS is a new method that obtained success to unconstrained continuous optimization and is being applied to some combinatorial optimization problems found in the literature [17]. The location-allocation algorithm application in the initial phase of ECS contributes to its improvement with competitive results to solved the PCTSP, getting to find the optimal solutions for instances up to 51 nodes. Besides, this approach obtained good results for the larger problems. These results validate the proposed approach and its use to solved the PCTSP.

Future works should research the effect of this location-allocation algorithm in an alternative approach, called *CS [4], which the evolutionary algorithm is substituted by alternative metaheuristics.

8. Acknowledgments:

The authors acknowledges Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq for partial financial research support.

References

[1] *ILOG CPLEX 7.5 Reference Manual*. ©Copyright by ILOG, France, 2001.

[2] E. Balas. The prize collecting travelling salesman problem. *Networks*, 19:621–636, 1989.

[3] A. A. Chaves, F. L. Biajoli, O. M. Mine, and M. J. F. Souza. Modelagens exata e heurística para resolução de uma generalização do problema do caixeiro viajante. *Simpósio Brasileiro de Pesquisa Operacional (SBPO)*, 36:1367–1378, 2004.

[4] A. A. Chaves and L. A. N. Lorena. Hybrid algorithms with detection of promising areas for the prize collecting travelling salesman problem. *Fifth international conference on Hybrid Intelligent Systems (HIS)*, pages 49–54, 2005.

[5] G. Croes. A method for solving travelling salesman problems. *Operations Research*, 6:791–812, 1958.

[6] M. Dell'Amico, F. Maffioli, and A. Sciomanchen. A lagrangian heuristic for the prize collecting travelling salesman problem. *Operations Research*, 81:289–305, 1998.

[7] D. Feillet, P. Dejax, and M. Gendreau. Travelling salesman problems with profits. *Transportation Science*, 2(39):188–205, 2005.

[8] T. Feo and M. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.

[9] M. Fischetti and P. Toth. An additive approach for the optimal solution of the prize collecting traveling salesman problem. *Vehicle Routing: Methods and Studies*, pages 319–343, 1988.

[10] F. Glover. Tabu search and adaptive memory programing: Advances, applications and challenges. *Interfaces in Computer Science and Operations Research*, pages 1–75, 1996.

[11] M. X. Goemans and D. P. Williamson. A general aproximation technique for constrained forest problems. *SIAM Journal on Computing*, 24(2):296–317, 1995.

[12] L. M. Gomes, V. B. Diniz, and C. A. Martinhon. An hybrid grasp+vnd metaheuristic fo the prize collecting traveling salesman problem. *Simpósio Brasileiro de pesquisa Operacional (SBPO)*, 32:1657–1665, 2000.

[13] L. A. N. Lorena and J. C. Furtado. Constructive genetic algorithm for clustering problems. *Evolutionary Computation*, 9(3):309–327, 2001.

[14] L. A. N. Lorena and M. A. Pereira. A lagrangean/surrogate heuristic for the maximal covering location problem using hillsman's edition. *International Journal of Industrial Engineering*, 9:57–67, 2002.

[15] V. A. Melo and C. A. Martinhon. Metaheurísticas híbridas para o problema do caixeiro viajante com coleta de prêmios. *Simpósio Brasileiro de Pesquisa Operacional (SBPO)*, 36:1295–1306, 2004.

[16] N. Mladenovic and P. Hansen. Variable neighborhood search. *Computers and Operations Research*, 24:1097–1100, 1997.

[17] A. C. M. Oliveira and L. A. N. Lorena. Detecting promising areas by evolutionary clustering search. *Advances in Artificial Intelligence. Springer Lecture Notes in Artificial Intelligence Series*, pages 385–394, 2004.

[18] A. C. M. Oliveira and L. A. N. Lorena. Population training heuristics. *Lecture Notes in Computer Science*, 3448:166–176, 2005.

[19] L. Shrage. *User's Manual for LINGO, LINDO Systems Inc*. Chicago, IL, 1991.

[20] G. Syswerda. Uniform crossover in genetic algorithms. *International Conference on Genetic Algorithms(ICGA)*, 3:2–9, 1989.

[21] R. D. Torres and J. A. M. Brito. Problemas de coleta de prêmios seletiva. *Simpósio Brasileiro de Pesquisa Operacional (SBPO)*, 35:1359–1371, 2003.