

根据该曲线可以看出:当交叉概率较小和较大时,解的质量较好;当交叉概率大于 0.8 以后,种群最优适应度的下降趋势不大.因此,可以设定 $p_c \geq 0.8$.

4.3 对比实验

为了验证本文所提出的混合遗传模拟退火算法(HGSA)在求解 SaaS 构件优化放置问题方面的有效性,我们又单独采用了遗传算法(GA)和模拟退火算法(SA)对该问题进行了求解.

在 GA 中,由于虚拟机的数量是不确定的,因此,通过交叉操作所产生的新染色体的质量不但没有改进反而在不断地下降.这是因为,交叉操作会增加虚拟机的数量,导致构件的放置更加稀疏,从而增加了放置方案的成本.为了提高 GA 的求解质量,我们采用本文所提出的新解产生规则(3)和规则(4)对交叉操作后所产生的新解进行了优化,通过实验分析发现其求解质量已显著提高.

传统的 SA 是从一个初始解开始不断进行邻域搜索来产生新解,因此,初始解的质量对算法的影响比较大.为了提高 SA 的求解质量,我们从初始种群中选择一个最优解作为 SA 的初始解.在传统的 SA 中,邻域搜索是通过改变染色体中某个基因的取值来实现的,该操作相当于本文所提出的邻域搜索规则(1),即,改变构件的放置位置.这一操作的收敛速度比较慢,而且很容易陷入局部最优/好解.为此,我们又将其他 3 个邻域搜索规则都加入到 SA 中,从而极大地提高了 SA 的求解质量.

另外,本文所提出的解初始化方法采用的是一种基于贪心策略的启发式算法(HA),即,每次放置构件都是尽量选择已经存在的虚拟机,这样可以减少虚拟机的使用数量.单独采用 HA 进行问题求解的运行效率比较高,但是解的质量较差.为了提高 HA 的求解质量,我们首先采用 HA 来生成一组解(种群),然后,从中选择一个最优解(相当于 SA 的初始解)作为 HA 的最终解.

本文通过实验对上述 4 种算法(HGSA,GA,SA 和 HA)的求解质量进行了比较.我们采用模拟的方式随机生成了 10 个构件模型,其构件数量分别是 10,20,...,100,每个构件的资源需求数量以及构件之间通信量按照第 4.1 节所给出的范围进行随机均匀生成.针对每一构件模型,我们分别采用上述 4 种算法对其进行求解 30 次取平均值.为了体现实验的公平性,每次求解它们都是从同一初始种群开始,其中,SA 的初始解是该种群中的最优解,HGSA 和 GA 的进化代数和交叉概率是一致的($s=100, p_c=0.9$),GA 的变异概率设置为 0.1.图 9 给出了 4 种算法所计算的总成本(虚拟机使用成本+网络通信成本)的对比分析图.

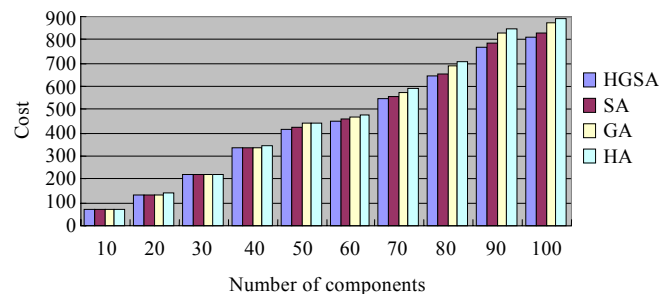


Fig.9 Comparison of total cost calculated by four algorithms

图 9 4 种算法所计算的总成本比较

通过实验分析发现:对于任意给定的初始种群,HGSA 求解质量最好,其次是 SA,再次是 GA,最后是 HA.由于 SA 是在 HA 的基础上继续进行邻域搜索,因此从理论上来说,SA 的求解质量至少不会比 HA 差.由于 HGSA 是采用多个解在其邻域内并行搜索的方式,并且利用交叉操作来提高全局搜索能力,因此其求解质量要高于 SA.与 HGSA 和 SA 相比,GA 对该问题的求解能力较弱,这主要是由于虚拟机数量的不确定性而导致搜索范围扩大,尽管我们希望通过相应的邻域搜索规则来改善新解的质量,但其效果仍不理想.

表 2 和表 3 分别给出了在每一构件数量下,4 种算法所计算的虚拟机使用成本和网络通信成本的具体数值.从实验结果可以看出:在每一个构件数量下,HGSA 所计算的虚拟机使用成本和网络通信成本都要低于其他 3

种算法;而且随着构件数量的增加,其求解质量显著提高.这说明 HGSA 在求解 SaaS 构件优化放置问题方面具有较好的效果.

Table 2 Comparison of virtual machine usage cost calculated by four algorithms

表 2 4 种算法所计算的虚拟机使用成本比较

	10	20	30	40	50	60	70	80	90	100
HGSA	67.53	135.48	214.84	322.27	399.38	423.93	501.14	587.31	695.33	713.64
SA	68.10	135.48	215.32	324.29	402.88	428.22	509.14	595.07	702.15	724.81
GA	67.53	135.48	215.32	325.34	415.60	434.91	521.45	622.67	741.77	757.01
HA	71.02	136.36	216.20	329.35	422.83	438.83	531.50	637.67	761.98	783.30

Table 3 Comparison of network communication cost calculated by four algorithms

表 3 4 种算法所计算的网络通信成本比较

	10	20	30	40	50	60	70	80	90	100
HGSA	0.35	0.97	4.41	10.33	18.92	29.45	43.09	56.55	75.76	94.49
SA	0.44	1.15	4.94	12.07	20.68	31.60	46.22	60.80	81.06	101.10
GA	0.35	1.22	5.71	13.13	22.41	33.98	49.12	64.80	86.11	106.62
HA	0.57	1.32	5.71	13.33	22.23	33.94	48.50	64.46	84.68	105.47

4.4 实例分析

本节通过图 1 所示的实例来说明 HGSA 在求解 SaaS 构件优化放置方面的能力.在该实例中,包含 10 个构件,每个构件的资源需求以及构件之间的通信如图 10 上半部分所示,虚拟机模板采用亚马逊的 EC2(具体配置见表 1).利用本文所提出的 HGSA 算法(算法参数设置为 $s=100, \alpha=0.9, T_{high}=250, T_{low}=0.01, l=6, p_c=0.9$),一次即可求得最优 SaaS 构件放置方案,如图 10 下半部分所示.在该方案中,我们选择了 2 台虚拟机,它们分别是 C3.xlarge 和 I2.2xlarge,其中,I2.2xlarge 上放置构件 $c_5, C3.xlarge$ 上放置其他所有构件,假设租期为 72h,网络通信价格为 0.01\$/GB,则最终成本为 67.602\$.其中,虚拟机使用成本为 67.536\$,网络通信成本为 0.066\$.这说明,本文所提出的算法在虚拟机数量不确定的情况下具有较好的效果.

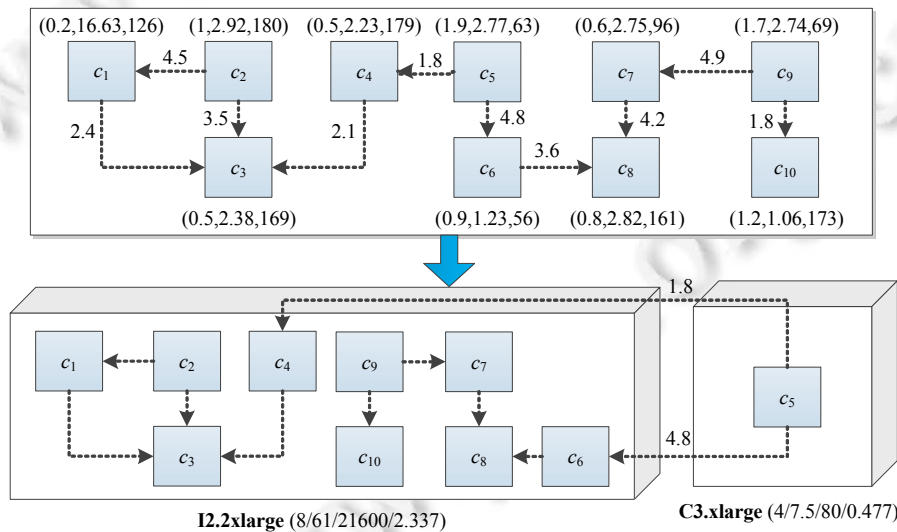


Fig.10 Texample of SaaS component optimal placement

图 10 SaaS 构件优化放置实例

5 结束语

针对传统软件提供商在向 SaaS 服务模式迁移时所遇到的 SaaS 构件放置问题,本文提出了一种基于混合遗

传模拟退火算法的 SaaS 构件优化放置方法,其优化目标是提高云资源利用率、降低云资源使用成本。尽管目前存在许多 SaaS 优化放置方法,但是这些方法都假定云环境中虚拟机的类型和数量是确定的,即在限定的资源上进行优化放置;然而实际情况是:虚拟机的类型和数量一般都是不确定的,需要根据被部署的构件资源需求来确定。而本文所提出的 SaaS 构件优化放置模型弥补了这些方法的不足。本文采用混合遗传模拟退火算法 HGSA 对 SaaS 构件优化放置问题进行求解,HGSA 算法结合了遗传算法和模拟退火算法的优点,克服了 GA 收敛速度慢和 SA 容易陷入局部最优的缺点,与单独使用遗传算法和模拟退火算法相比,HGSA 算法具有更高的求解质量。从云架构角度来看,本文所提出的放置方法不仅适用于公有云基础设施的放置,同时也适用于私有云基础设施的放置。从软件生命周期阶段来看,本文所提出的放置方法不仅适用于 SaaS 的初始部署,而且也适用于 SaaS 运行阶段的动态部署。

References:

- [1] Kang S, Kang S, Hur S. A design of the conceptual architecture for a multitenant SaaS application platform. In: Proc. of the Int'l Conf. on Computers, Networks, Systems, and Industrial Engineering. IEEE Computer Society Press, 2011. 462–467. [doi: 10.1109/CNSI.2011.56]
- [2] Zhang Y, Wang ZH, Gao B, Guo CJ, Sun W, Li XP. An effective heuristic for on-line tenant placement problem in SaaS. In: Proc. of the 2010 IEEE Int'l Conf. on Web Services. IEEE Computer Society Press, 2010. 425–432. [doi: 10.1109/ICWS.2010.65]
- [3] Yu HY, Wang DH. System resource allocation algorithm for multi-tenant SaaS application. In: Proc of the 2011 Int'l Conf. on Cloud and Service Computing. IEEE Computer Society Press, 2011. 207–211. [doi: 10.1109/CSC.2011.6138523]
- [4] Wu LL, Garg SK, Buyya R. SLA-Based admission control for a software-as-a-service provider in cloud computing environments. Journal of Computer and System Sciences, 2012,78(5):1280–1299. [doi: 10.1016/j.jcss.2011.12.014]
- [5] Yang EF, Zhang Y, Wu L, Liu YL, Liu SJ. A hybrid approach to placement of tenants for service-based multi-tenant SaaS application. In: Proc. of the 2011 IEEE Asia-Pacific Services Computing Conf. IEEE Computer Society Press, 2011. 124–130. [doi: 10.1109/APSCC.2011.35]
- [6] Tian C, Jiang HB, Iyengar A, Liu X, Wu ZD, Chen JH, Liu WY, Wang CG. Improving application placement for cluster-based Web applications. IEEE Trans. on Network and Service Management, 2011,8(2):104–115. [doi: 10.1109/TNSM.2011.050311.100040]
- [7] Yusoh ZIM, Tang M. Composite SaaS placement and resource optimization in cloud computing using evolutionary algorithms. In: Proc. of the 2012 IEEE 5th Int'l Conf. on the Cloud Computing. IEEE Computer Society Press, 2012. 590–597. [doi: 10.1109/CLOUD.2012.61]
- [8] Lloyd W, Pallickara S, David O, LyonbAuthor J, ArabibAuthor M, Rojas K. Performance implications of multi-tier application deployments on infrastructure-as-a-service clouds: Towards performance modeling. Future Generation Computer Systems, 2013, 29(5):1254–1264. [doi: 10.1016/j.future.2012.12.007]
- [9] Moens H, Truyen E, Walraven S, Joosen W, Dhoedt B, De Turck F. Cost-Effective feature placement of customizable multi-tenant application in the cloud. Journal of Network System Management, 2014,22(4):517–588. [doi: 10.1007/s10922-013-9265-5]
- [10] Zhu XY, Santos C, Beyer D, Ward J, Singhal S. Automated application component placement in data centers using mathematical programming. Int'l Journal of Network Management, 2008,18:467–483. [doi: 10.1002/nem.707]
- [11] Jin ZH, Cao J, Li ML. A distributed application component placement approach for cloud computing environment. In: Proc. of the 9th IEEE Int'l Conf. on Dependable, Automic and Secure Computing. IEEE Computer Society Press, 2011. 488–495. [doi: 10.1109/DASC.2011.94]
- [12] Sailer A, Head MR, Kochut A, Shaikh H. Graph-Based cloud service placement. In: Proc. of the 2010 IEEE Int'l Conf. on Services Computing. IEEE Computer Society Press, 2010. 89–96. [doi: 10.1109/SCC.2010.67]
- [13] Pisinger D. An exact algorithm for large multiple knapsack problem. European Journal of Operational Research, 1999,114(3): 528–541. [doi: 10.1016/S0377-2217(98)00120-9]
- [14] Kataoka S, Yamada T. Upper and lower bounding procedures for the multiple knapsack assignment problem. European Journal of Operational Research, 2014,237(2):440–447. [doi: 10.1016/j.ejor.2014.02.014]

- [15] Sarac T, Sipahioglu A. Generalized quadratic multiple knapsack problem and two solution approaches. *Computer & Operations Research*, 2014,43:78–89. [doi: 10.1016/j.cor.2013.08.018]
- [16] Ren ZG, Feng ZR, Zhang AM. Fusing ant colony optimization with Lagrangian relaxation for the multiple-choice multidimensional knapsack problem. *Information Science*, 2012,182(1):15–29. [doi: 10.1016/j.ins.2011.07.033]
- [17] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. on Evolutionary Computation*, 2002,6(2):182–197. [doi: 10.1109/4235.996017]
- [18] McCall J. Genetic algorithms for modelling and optimization. *Journal of Computational and Applied Mathematics*, 2005,184(1):205–222. [doi: 10.1016/j.cam.2004.07.034]
- [19] Chen D, Lee CY, Park CH. Hybrid genetic algorithm and simulated annealing (HGASA) in global function optimization. In: *Proc. of the 17th IEEE Int'l Conf. on Tools with Artificial Intelligence*. IEEE Computer Society Press, 2005. 113–117. [doi: 10.1109/ICTAI.2005.72]
- [20] Li WD, Ong SK, Nee AYC. Hybrid genetic algorithm and simulated annealing approach for the optimization of process plans for prismatic parts. *Int'l Journal of Production Research*, 2002,40(8):1899–1922. [doi:10.1080/00207540110119991]
- [21] Loukil L, Mehdi M, Mebab N. A parallel hybrid genetic algorithm-simulated annealing for solving Q3AP on computational grid. In: *Proc. of the IEEE Int'l Symp. on Parallel & Distributed Processing*. IEEE Computer Society Press, 2009. 1–8. [doi: 10.1109/IPDPS.2009.5161126]
- [22] Li YH, Guo H, Wang L, Fu J. A hybrid genetic-simulated annealing algorithm for the location-inventory-routing problem considering returns under E-supply chain environment. *The Scientific World Journal*, 2013. [doi: 10.1155/2013/125893]



孟凡超(1974—),男,黑龙江依兰人,博士,副教授,CCF 高级会员,主要研究领域为软件工程,软件体系结构,云计算,服务计算。



李克秋(1971—),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为网络,云计算,数据中心。



初佃辉(1969—),男,博士,教授,CCF 高级会员,主要研究领域为服务计算与服务工程,面向服务的软件体系结构。



周学权(1979—),男,讲师,主要研究领域为云计算,SaaS。