

P2P 网络服务器部署方案及其启发式优化算法*

曾明霏, 余顺争

(中山大学 信息科学与技术学院, 广东 广州 510006)

通讯作者: 余顺争, E-mail: syu@mail.sysu.edu.cn

摘要: 为了更好地利用服务器提升 P2P 网络的性能, 提出了一种 P2P 网络服务器最优化部署方案. 该方案将如何部署有限的服务器资源来满足尽量多节点的问题, 转化为一个带有约束条件的最优化问题. 考虑到极坐标空间适合于表达长度相关的约束条件, 例如部署方案中的约束条件, 提出一种基于极坐标映射的启发式算法, 解决 P2P 网络服务器最优部署问题. 相比于通常的使用罚函数的启发式算法, 该算法可以有效地减小搜索空间, 其性能的改善在搜索空间维数高的情况下更加显著. 实验结果表明, 所提出的服务器部署方案可以满足更多节点的需求, 能够有效提高 P2P 网络的性能.

关键词: P2P 网络; 服务器部署; 启发式算法; 极坐标空间; 约束条件

中图法分类号: TP393 **文献标识码:** A

中文引用格式: 曾明霏, 余顺争. P2P 网络服务器部署方案及其启发式优化算法. 软件学报, 2013, 24(9): 2226-2237. <http://www.jos.org.cn/1000-9825/4362.htm>

英文引用格式: Zeng MF, Yu SZ. Server allocation approach for P2P networks and its heuristic algorithm. Ruan Jian Xue Bao/ Journal of Software, 2013, 24(9): 2226-2237 (in Chinese). <http://www.jos.org.cn/1000-9825/4362.htm>

Server Allocation Approach for P2P Networks and its Heuristic Algorithm

ZENG Ming-Fei, YU Shun-Zheng

(School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510006, China)

Corresponding author: YU Shun-Zheng, E-mail: syu@mail.sysu.edu.cn

Abstract: An optimal allocation approach of servers for improving the performance of P2P networks is proposed in this paper. The approach is to allocate a limited number of servers over the P2P networks to satisfy as many peers as possible, by using a constrained optimization algorithm. In considering that the polar space is more suitable to express length-related constraints, such as those in this approach, a heuristic algorithm based on the polar coordinator mapping is proposed to deal with the optimization problem of the server allocation. Compared with the existing heuristic algorithms that are based on penalty functions, this algorithm can effectively reduce the search space, and the performance is better especially in a high dimensional search space. The experiment results show that our server allocation approach can satisfy more peers' requirement and effectively improve the performance of P2P networks.

Key words: P2P network; server allocation; heuristic algorithm; polar space; constraint

P2P 网络的不同节点之间可以直接交换数据或者提供服务, 具有开放、健壮、灵活等特性, 能够满足节点各种不同的需求. 由于 P2P 网络的节点数量巨大和 P2P 网络的优势, 越来越多的公司或者组织将服务器投入 P2P 网络之中, 利用 P2P 网络进行大规模内容分发、视频点播等服务. 例如, 在 IPTV^[1]网络和 CDN-P2P^[2]网络中, 终端用户和面向终端用户的服务器组成一个 P2P 网络, 利用 P2P 网络的特性提高整个网络的性能.

这种带有服务器的 P2P 网络与普通的无服务器的 P2P 网络一样, 网络中的用户(peers, 后文称节点)既是内容的请求者, 又是内容的提供者. 不同的是, 其中的服务器只是内容的提供者而不是内容的请求者. 由于有服务

* 基金项目: 国家自然科学基金(60970146); 国家高技术研究发展计划(863)(2007AA01Z449)

收稿时间: 2012-04-27; 修改时间: 2012-10-19; 定稿时间: 2012-12-27

器提供大量的内容等资源,节点在下载完成后有很大的概率离开网络(或失效)^[3],因此,这类带有服务器的 P2P 网络只有在服务器的持续激励下才能保持稳定,服务器依然是网络的核心.显然,服务器拥有的带宽和内容越多,能激励的节点越多,节点的下载速度越快.下载速度快的 P2P 网络自然会受到节点欢迎,节点也会越来越多.

由于网络拓扑的影响,不同地区之间的延迟较大而且带宽较小,P2P 网络的节点倾向于和同一地区的节点交换资源,这就造成不同地区的 P2P 网络存在一定的差别.在带有服务器的 P2P 网络中,这种地区差别更明显,因为服务器不仅要考虑延迟和带宽的问题,还需要考虑跨 ISP 流量的问题.所以,带有服务器的 P2P 网络往往根据网络拓扑划分为若干区域.服务器管理者为了吸引节点加入 P2P 网络,会在每个区域设置区域服务器,区域服务器只为区域中的节点提供服务.规模较大的 P2P 网络还会部署一些跨区域服务器同步这些区域服务器提供的内容,进一步提高服务的质量.

这种 P2P 网络可以通过增加服务器获得更好的性能,而且根据 IPTV 网络的研究表明,现有 IPTV 网络的服务器增加很多也没有出现过剩的情况^[4].但是从投入成本考虑,在网络中部署的服务器数量和容量一定是有限的,需要在考虑地区差别的情况下进行最优部署.

本文为解决上述问题,提出一种服务器部署方案,它按照地区差别进行服务器部署,并将部分服务器作为共享服务器部署在网络中可跨区域的位置,通过动态分配这些共享服务器资源来改善网络的性能.共享服务器的动态分配在数学上是一个带约束条件的最优化问题.为解决该问题,本文提出一种基于极坐标映射的启发式算法.这种算法在解决本文的约束条件最优化问题时有很好的效果.实验结果表明,基于极坐标映射的启发式算法比普通的启发式算法的效果好,在高维空间中的性能改善更加明显.基于极坐标映射的启发式算法的 P2P 网络服务器最优部署方案可以满足更多节点的需求,能够有效提高 P2P 网络的吸引力.

本文第 1 节是带有服务器的 P2P 网络和启发式算法的相关研究.第 2 节是 P2P 网络服务器部署问题模型.第 3 节是基于极坐标映射的启发式算法.第 4 节是基于极坐标映射的启发式算法的性能实验.第 5 节是关于本文 P2P 网络服务器部署方案的实验.第 6 节是本文的总结.

1 相关研究

带有服务器的 P2P 网络架构主要有 IPTV 网络和 CDN-P2P 网络.IPTV^[1]网络的视频源服务器通过 P2P 架构向外发布视频流,与传统的 C/S 架构相比,同样的视频源服务器可以为更多用户提供服务.所以,这是最常见的带有服务器的 P2P 网络.CDN-P2P^[2]网络是在内容发布网 CDN 网络架构中加入 P2P 技术,也是一种混合架构的网络.在这种类型的网络中,CDN 核心服务器和边界服务器依然使用传统的 CDN 网络内容推送机制,而在边界服务器和终端用户之间使用 P2P 架构进行内容传输,因而有效地提高了 CDN 网络的性能.现有研究表明,由于应用和用户的分布式特点,网络的工作负载存在地区差异性.

P2P 中服务器部署策略问题,实际上就是一个约束最优化问题.现有许多约束最优化算法,其中,遗传算法和粒子群算法获得了广泛的应用.遗传算法通过模拟生物的自然选择和基因的交叉变异过程解决最优化问题^[5],它不需要额外信息就可以从搜索空间中找到性能优秀的可行解,是一种解决优化问题的通用算法.不过,遗传算法在迭代过程中并没有储存任何启发信息,所以对于不少复杂问题,遗传算法虽然可以收敛,但是收敛的速度较慢.粒子群算法^[6,7]是在遗传算法之后提出出来的一种启发式算法,其通过模拟鸟群飞行、鱼群游弋和虫群觅食等行为,从搜索空间中寻找可行解.与遗传算法相比,粒子群算法的启发性更强,它可以储存在搜索过程中遇到的最优解,利用这些信息加速收敛的过程.但是,粒子群算法也比遗传算法更容易陷入局部极值.

结合遗传算法和粒子群算法的混合算法取得了更好的效果^[8-10],它们使用粒子群算法进行局部搜索加快收敛速度,同时使用遗传算法避免局部极值.

无论粒子群算法还是遗传算法,都可以应用在极坐标空间中^[11,12].但是对于一般的无约束条件最优化问题,极坐标空间并没有优势.

2 具有服务器的 P2P 网络服务器部署问题

2.1 具有服务器的 P2P 网络结构

由于存在跨 ISP 流量的问题,带有服务器的 P2P 网络比普通 P2P 网络的地区差异性更大,网络中的服务器或者节点都极少为不在同一地区的节点提供服务.因此,本文假设 P2P 网络依据地区不同划分为若干区域,每个区域的区域级服务器只为本区域的节点提供服务.

如果某个区域突然新增了大量节点,区域内的服务器无法满足激增的节点需求,而其他区域的服务器因为区域差异性的问题难以协助,这会导致节点体验质量下降,进而流失节点.为了解决此类问题,服务器管理者可以设立全局级服务器,把它们部署在骨干网上,为 P2P 网络提供全局服务.整个网络的结构如图 1 所示.

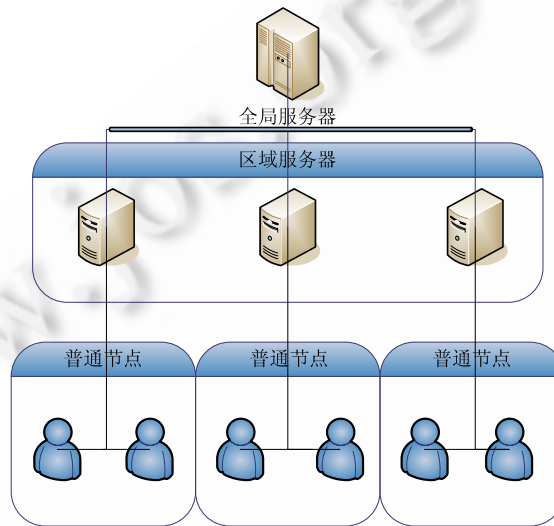


Fig.1 Structure of P2P network with servers

图 1 带有服务器的 P2P 网络的网络结构

2.2 搭便车节点数量与系统的平衡点

假设处于正常状态的理性节点除了从网络中下载(获取)以外,还会主动上传(贡献);而处于搭便车状态的节点只下载不上传;服务器则只上传不下载.网络中所有节点都是根据贡献与获取比例的大小理性地决定自己的行为.令 x_i 表示当前的负担,即贡献与获取的比例, $\delta_i \geq 0$ 表示节点 i 能承受的最大负担,则节点 i 的行为模型如下:

$$\begin{cases} \text{normal,} & \text{if } x_i < \delta_i \\ \text{free-rider,} & \text{otherwise} \end{cases}$$

即, δ_i 小的理性节点更容易转换成搭便车状态.

负担 x_i 的大小和处于搭便车状态的节点数量有关.假定节点在一个时间段内不会改变状态,只会从在一个时间段到下一个时间段转换的时刻才会转换状态.在时间段 t 结束之前,令 p_t 表示当前区域中节点的数量, f_t 表示搭便车节点的比例, s_t 表示区域中服务器的数量.设 s_t 是一个可调节的量,用于对网络整体状况的控制,因而可以假定 s_t 与 p_t 成正比,即 $s_t = \rho p_t$. 则 αp_t 就是对资源的总需求,其中, α 代表平均每个节点的需求.该总需求将由服务器和所有理性节点共同承担,即

$$x_i = \frac{\alpha p_t}{\beta p_t(1-f_t) + \gamma s_t} = \frac{\alpha}{\beta(1-f_t) + \gamma \rho} \quad (1)$$

其中, β 代表平均每个理性节点的贡献, γ 代表平均每个服务器的贡献.可见:

$$\frac{\alpha}{\beta + \gamma\rho} \leq x_t \leq \frac{\alpha}{\gamma\rho} \tag{2}$$

假设 δ_i 满足区间 $[0, \delta_{\max}]$ 的均匀分布, 在负担为 x_t 的情况下, 所有 $\delta_i \leq x_t$ 的理性节点都会在下一个时间段 $t+1$ 转换成搭便车状态, 即 $f_{t+1} = \Pr(\delta_i \leq x_t) = x_t / \delta_{\max}$. 由于开放的 P2P 网络中搭便车现象不可能消除^[13], 网络中始终会存在少量处于搭便车状态的节点, 令 f_{\min} 表示搭便车节点比例的下限, 则 $f_{\min} = \Pr(\delta_i \leq \alpha / (\beta + \gamma\rho))$. 图 2 是 x_t 和 f_t 的关系图, 图中的两条曲线分别为公式(1)和 $f_{t+1} = x_t / \delta_{\max}$.

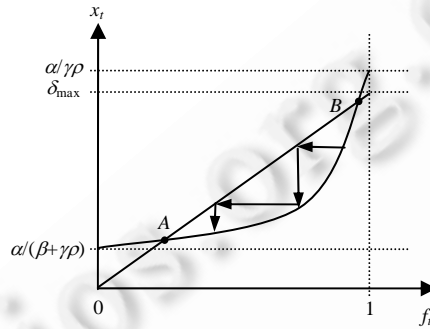


Fig.2 Relation between x_t and f_t

图 2 负担 x_t 和搭便车节点 f_t 比例关系图

这两条曲线的交点 A 和 B 为两个系统平衡点, 当系统处于这两点时, 搭便车节点比例正好和负担相匹配, 不会发生变化. 当系统的状态处于 A 和 B 之间时, 较小的负担会使得系统中搭便车节点比例下降, 搭便车节点比例下降会进一步减轻负担, 系统状态就会像图 2 中的箭头那样逐步趋向于 A 点. 当系统的状态处于 $0 \sim A$ 之间时, 少量搭便车节点造成的负担就会使得搭便车节点比例上升, 并且进一步加大负担, 这会导致系统状态趋向于 A 点. 而当系统的状态处于 $B \sim 1$ 之间时, 系统的负担太高, 搭便车节点比例会趋向于 1, 即所有节点都变成了搭便车节点.

显然, A 是稳定的平衡点, B 是不稳定的平衡点. 在 A 点附近的系统状态都会在系统的作用下趋近 A 点; 而在 B 点附近的系统状态则有可能趋近到 $f_t = 1$ 的点. 要避免这种情况, 系统必须加大服务器的投入比例 ρ , 使得 $(\alpha / \gamma\rho) \leq \delta_{\max}$, 即令 $\rho \geq \alpha / (\gamma\delta_{\max})$. 在这种情况下, 系统只有一个平衡点 A , 它稳定在少的搭便车节点数和小的平均负担这种良好状态.

2.3 服务器投入比例与节点的加入和离开

由上一节的分析可知, 通过调节服务器在某个区域的投入比例 ρ , 可以调节该区域的系统状态. 处于良好状态的 P2P 系统除了可以加速节点的下载速度, 还可以吸引更多的节点加入. 在稳态情况下, 节点的到达率和下载完成后的离开率应该是相等的. 如果有更多的节点加入, 则会导致该区域的系统总体负担的增加, 有可能使得系统状态变坏. 这时, 只有加大服务器在该区域的投入比例 ρ 以改进其系统状态, 才能保留住新节点.

因此, 从另一个角度看, 加大服务器的投入比例 ρ 才可以吸引更多的新节点加入一个区域. 令 $\rho_t = s_t / p_t$ 代表在第 t 个时间段时服务器的投入比例, 因此可以假定, 新增节点的数量与 $\rho_t - \alpha / (\gamma\delta_{\max})$ 的大小成正比, 比例系数为常数 c , 与网络目标规模有关. 综合考虑公式(1), 可以得到公式(3).

$$\left. \begin{aligned} p_{t+1} &= p_t + c \left(\rho_t - \frac{\alpha}{\gamma\delta_{\max}} \right) \\ x_{t+1} &= \frac{\alpha}{\beta(1 - f_{t+1}) + \gamma\rho_{t+1}} \\ f_{t+1} &= x_t / \delta_{\max} \end{aligned} \right\} \tag{3}$$

如果在随后的 k 个时间段内保持服务器的数量 s_i 不变,则随着 $k \geq 1$ 的增加, ρ_{t+k} 变小,直到 $\rho_{t+k} = \alpha / (\gamma \delta_{\max})$ 后,节点总数不再增加,该区域的 P2P 系统进入一个新的稳定状态.即,稳定状态的节点总数由投入的服务器数量决定,服务器越多,节点数越多.

2.4 服务器部署问题

带有服务器的 P2P 网络的客户端一般是由服务器管理者定制的,因此,服务器管理者可以通过定制的客户终端获取不同区域中的节点数.服务器管理者还可以通过抽样调查等方式估计区域的其他状态.例如,BT 协议可以要求客户端向 Tracker 汇报当前的上传速度,Tracker 从这些数据中可以估算搭便车节点的比例.基于上述状态数据,服务器管理者可以根据公式(3)推测未来一段时间中每个区域的状态,并根据推测合理地调节各个区域的服务器数量.

在服务器分区级和全局级两级部署的情况下,区域级服务器只能为对应区域服务,不具备调节的能力;而全局级服务器可以为整个网络服务,具有调节能力.服务器管理者可以通过修改设置使得全局级服务器专门为某个区域服务,这相当于为某个区域临时部署了一些服务器.因此,投入区域 i 的服务器数量 $s_i^{(i)}$ 是专门用于区域 i 的服务器数量 $r(i)$ 与为区域 i 部署的全局服务器数量 $g_i^{(i)}$ 的和,即 $s_i^{(i)} = r(i) + g_i^{(i)}$.

区域级服务器的数量一般不会改变,可视作常数.全局级服务器在部署完成之后会持续为相应区域提供服务,在 k 个时间段后,管理者再次进行全局级服务器的分配.因此,在全局级服务器部署完成后的 k 个时间段内,某个区域投入的服务器数量不会改变. k 个时间段结束后,区域中的节点数 p_{t+k} 和搭便车节点比例 f_{t+k} 的推测值由全局服务器的部署方案决定.

由公式(3)可知,更多的服务器可以吸引更多的正常节点,而服务器管理者可进行分配的全局级服务器数量有限,很可能无法同时消除所有区域的不稳定点 B .如何分配这些有限的全局级服务器,使得 k 个时间段后所有区域的正常节点总数尽量多,是一个复杂的优化问题.令 G 表示全局级服务器总数, n 表示区域数量,则 P2P 网络服务器部署问题可以描述如下:

$$\left. \begin{array}{l} \max_{(g_{t+k}^{(1)}, \dots, g_{t+k}^{(n)})} \sum_{i=1}^n p_{t+k}^{(i)} (1 - f_{t+k}^{(i)}) \\ \text{s.t.} \quad \sum_{i=1}^n g_{t+k}^{(i)} \leq G \end{array} \right\} \quad (4)$$

其中, $g_{t+k}^{(i)} \geq 0$ for all i ,且每个区域的 p_{t+k} 和 f_{t+k} 可以通过公式(3)计算.

由于重新部署服务器的时间间隔的大小会影响优化的效果,间隔时间过长会导致服务器部署方案无法适应动态变化的 P2P 网络,间隔时间过短则会增加重新部署服务器的代价,所以重新部署服务器的时间间隔的选取应该是在 P2P 网络状态出现节点数下降、搭便车节点比例增加较明显时,即明显偏离公式(3)的稳定状态时进行,并且给重新部署服务器的时间间隔设置一个上限和下限.例如,至少一周重新部署一次、最多一天重新部署一次.

3 基于极坐标映射的启发式算法

3.1 约束条件与极坐标映射

优化问题(4)是一个带有约束条件的复杂优化问题,它可以使用启发式算法进行求解.如果令 $x_i^2 = g_{t+k}^{(i)}$,其中, $g_{t+k}^{(i)} \geq 0$, $t+k$ 对于给定的时间段是不变的, i 代表拟分配服务器资源的网络区域,则其约束条件具有公式(5)所示的形式:

$$\left. \begin{array}{l} h(x) = \sum_{i=1}^n x_i^2 - r^2 = 0 \\ g(x) = \sum_{i=1}^n x_i^2 \leq r^2 \end{array} \right\} \quad (5)$$

显然,满足公式(5)中 $h(x)$ 的空间是一个 n 维空间中的超球面,满足 $g(x)$ 的空间是一个超球体.这两个约束条

件通常存在于一些与长度相关的最优化问题中,并且很适于在极坐标空间中表示.因此,我们使用极坐标映射处理该约束问题.

笛卡尔空间到极坐标空间的映射公式如下:

$$\left. \begin{aligned} \theta_j &= \cos^{-1}(x_{n-j+1}/(r \sin \theta_1 \dots \sin \theta_{j-1})), j=1, \dots, n-2 \\ \cos \theta_{n-1} &= x_1 / (r \sin \theta_1 \dots \sin \theta_{n-2}) \\ \sin \theta_{n-1} &= x_2 / (r \sin \theta_1 \dots \sin \theta_{n-2}) \end{aligned} \right\} \quad (6)$$

其中, $r = \sqrt{x_1^2 + \dots + x_n^2}$. 逆映射公式如下:

$$x_i = r \sin \theta_1 \dots \sin \theta_{n-i} \cos \theta_{n-i+1}, i=1, \dots, n \quad (7)$$

其中, $\theta_n=0$. 公式(6)和公式(7)都是一对一映射,都具有连续性. 笛卡尔空间下的任意一点有且只有一个极坐标空间下的点与之对应,反之亦然.

对于约束条件公式(5)所示的 n 维空间中的超球面 $h(x)$, 根据极坐标映射公式,可行解域映射到极坐标空间中之后,半径 r 是固定的,而 $n-1$ 维角度是自由的. 这使得启发式算法在极坐标空间中变得更加简单,进而可以加速算法的计算速度. 特别的,如果 $h(x)$ 是最优化问题的唯一约束条件,那么在极坐标空间中,这个带约束条件的问题可以转化为无约束条件的问题,极大地提高了启发式算法的性能.

类似于 $h(x)$, 约束条件超球体 $g(x)$ 在极坐标空间中可以通过将半径限制在区间 $[0, r]$ 内来满足. 所以在极坐标空间中,约束条件 $g(x)$ 带来的限制仅仅存在于半径这一维,而其他维度是自由的. 因此,对于约束条件 $g(x)$, 在极坐标空间下的启发式算法依然比笛卡尔空间的要高效.

当空间的维数提高时,极坐标空间的优势会更加明显. 因为在高维空间中,笛卡尔空间的搜索违反约束条件的概率会越大,而极坐标空间的搜索不会受到空间维数的影响,可以保持相同的搜索效率.

3.2 极坐标空间边界问题与初始化问题

边界问题是粒子群算法里的一个重要问题,其他启发式算法中也可能存在类似的问题. 在极坐标空间中,边界问题由两个部分组成:半径问题和角度问题^[14]. 极坐标映射及其逆映射都是由三角函数构成. 三角函数都是周期性的连续函数,周期都为 2π . 因此,公式(7)所示的逆极坐标映射也应该具有周期 2π . 由此可以推导得到一个极坐标空间中的一个周期性映射方法,如,令 $\mu(r, \vec{\theta})$ 表示对坐标 $(r, \vec{\theta})$ 进行逆极坐标映射,则周期性映射方法可表示为公式(8)和公式(9).

$$\mu(r, \theta_1, \theta_2, \dots, \theta_i, \dots, \theta_{n-1}) = \mu(r, \theta_1, \dots, \theta_i \pm 2\pi, \dots, \theta_{n-1}), i=1, 2, \dots, n-1 \quad (8)$$

$$\mu(r, \theta_1, \theta_2, \dots, \theta_i, \dots, \theta_{n-1}) = \mu(r, \theta_1, \dots, 2\pi - \theta_i, \theta_{i+1} + \pi, \dots, \theta_{n-1}), i=1, 2, \dots, n-1 \quad (9)$$

任意的一组角度都可以通过这个方法映射得到且只能得到一组符合边界条件的角度. 这个周期性映射在粒子群算法更新粒子速度和位置向量后使用,可以让粒子在进化的过程中忽略边界的影响,提升粒子群算法的性能.

结合半径上的边界问题,极坐标空间下边界问题处理算法的流程如下:

算法 1. 极坐标空间下边界问题处理算法.

$pv=(r, \theta_1, \theta_2, \dots, \theta_{n-1})$ 是任意极坐标空间下的位置向量

If $r > r_{\max}$ Then $r = r_{\max}$;

If $r < 0$ Then $r = 0$;

For $i=1$ to $n-2$

$\theta_i = \theta_i \bmod 2\pi$;

If $\theta_i > \pi$

$\theta_i = 2\pi - \theta_i$;

$\theta_{i+1} = \theta_{i+1} + \pi$;

EndIf

EndFor

$\theta_{n-1} = \theta_{n-1} \bmod 2\pi;$

极坐标空间初始化方法是在生成初始种群时将半径固定为 r , 角度在边界范围内随机产生, 生成出来的种群都会符合约束条件 $h(x)$, 并且在半径为 r 的超球面上均匀分布. 如果半径在 $[0, r]$ 范围内随机产生, 角度在边界范围内随机产生, 那么所有种群都将满足约束条件 $g(x)$, 并且在半径为 r 的超球体内部均匀分布.

3.3 极坐标交叉算子

在遗传算法中, 交叉算子是一种用于产生下一代种群染色体编码的方法. 它通过模拟自然界的染色体交叉复制过程, 将两个亲代的染色体编码通过一定的方式合成得到子代的染色体编码.

在实数编码方案并且在搜索空间连续的情况下, 加权平均可以看作是一种学习的过程. 交叉算子的目的是为了从其他个体中学习更好的基因编码, 因此, 加权平均在极坐标空间中可以作为一种特殊的交叉算子, 它不会有传统交叉算子那样对很小的染色体不适用的情况. 本文使用的交叉算子见公式(10).

$$\left. \begin{aligned} \text{parent}_1 &= (r_1, \bar{\theta}_1), \text{parent}_2 = (r_2, \bar{\theta}_2) \\ \text{child} &= (k \times r_1 + (1-k) \times r_2, k \times \bar{\theta}_1 + (1-k) \times \bar{\theta}_2) \end{aligned} \right\} \quad (10)$$

其中, $k \in [0, 1]$ 是一个随机变量.

3.4 基于极坐标映射的启发式算法及其复杂度分析

基于极坐标映射的启发式算法是在普通启发式算法中加入极坐标映射, 使得启发式算法能在极坐标空间中进行迭代搜索, 具体包括以下 5 个步骤:

- (1) 在极坐标空间产生初始种群, 或者在笛卡尔空间中产生初始种群后, 使用公式(6)映射到极坐标空间;
- (2) 将优化问题的目标函数改写为极坐标空间下的目标函数, 或者在计算目标函数前使用公式(7)变换坐标系;
- (3) 粒子群算法的速度向量和位置向量的更新方法可以直接用于极坐标空间中, 空间的边界问题需用算法 1 解决;
- (4) 遗传算法的交叉算子使用公式(10), 变异不能突破极坐标边界约束;
- (5) 在输出最后结果前使用公式(7)变换坐标系.

基于极坐标映射的遗传算法或者粒子群算法, 与原有算法的主要区别是增加了极坐标映射. 本文采用的算法是一种遗传和粒子群混合的 HGAPSO 算法, 其时间和空间复杂度见文献[10]. 而有关极坐标映射带来的算法复杂度分析如下:

根据公式(7), 一次 n 维极坐标映射需要进行 $O(n)$ 次三角函数计算及进行 $O(n^2)$ 次乘法运算. 在对精度要求不是特别高的情况下, 三角函数的计算可以通过查表和简单插值进行, 即只需要常数次查表和常数次乘法运算就可以完成一次三角函数的计算. 所以极坐标映射的时间复杂度为 $O(n^2)$ 次乘法运算, 不是特别高. 在空间复杂度上, 通过查表、插值的方式计算三角函数需要一个三角函数表, 三角函数表的大小与需要的精度有关. 对精度的要求越高, 三角函数表越大. 将精度控制在合理范围内, 空间复杂度也不高.

4 基于极坐标映射的启发式算法的性能评估实验

本文选择对一种遗传和粒子群混合的 HGAPSO 算法进行极坐标映射.

HGAPSO 算法在遗传算法部分使用实数编码, 并且个体的染色体编码就是粒子的位置向量. 在每次迭代中, 种群首先由粒子群算法增强, 然后进行自然选择、交叉变异得到新一代种群. 该算法在保留粒子群算法收敛迅速的特点的同时, 吸收了遗传算法不易陷入局部极值的特性, 具有比普通粒子群和遗传算法更好的效果.

根据初始化方式和搜索迭代的空间的不同, 对 HGAPSO 算法进行的极坐标映射有 4 种方式, 见表 1, 性能评估实验对这 4 种情况进行评估. 表中 S1 表示原始的 HGAPSO 算法; S2 只在极坐标空间产生初始种群, 其他部分为原始的 HGAPSO 算法; S3 在产生初始种群时使用原始算法, 但在极坐标空间进行迭代搜索; S4 将 HGAPSO 算

法完全改造为基于极坐标映射的启发式算法.性能评估实验中用到的其他参数全部参照文献[10]选取.

Table 1 Settings of benchmark

表 1 性能评估实验的设置

评估设定	初始化空间	迭代搜索空间
S1	Cartesian	Cartesian
S2	Polar	Cartesian
S3	Cartesian	Polar
S4	Polar	Polar

4.1 性能评估函数

在实验中使用的性能评估函数来自文献[15].

第 1 个性能评估函数称为 G3 问题,它拥有一个约束条件:

$$\left. \begin{aligned} \max_x f(x) &= (\sqrt{n})^n \prod_{i=1}^n x_i \\ \text{s.t. } g(x) &= \sum_{i=1}^n x_i^2 \leq 1 \\ \text{The global minima: } x^* &= (1/n^{0.5}, \dots, 1/n^{0.5}), f(x^*) = 1 \end{aligned} \right\} \quad (11)$$

在实验中,目标函数的维数 n 分别设置为 10,40,70 和 100.初始种群中包含 400 个个体,迭代进行 500 代.

在极坐标空间初始化时,个体的半径在 [0,1] 的范围内随机产生;在笛卡尔空间初始化时,个体在超立方体 $x_i \in [-1,1], i=1,2, \dots, n$ 中随机产生.S2 需要使用公式(7)将初始种群从极坐标空间转换到笛卡尔空间,S3 需要使用公式(6)进行初始种群的转换.

当搜索在极坐标空间进行时,迭代产生的个体均满足约束条件,故不需要罚函数,适应度函数直接使用目标函数.笛卡尔空间不具备上述特性,必须设置罚函数.公式(12)是该问题使用的适应度函数,它可以使得离可行解域越远的个体适应度越低.

$$\text{fitness} = \begin{cases} f(x), & \text{if the individual is feasible} \\ -\sum_{i=1}^n x_i^2, & \text{otherwise} \end{cases} \quad (12)$$

第 2 个性能评估函数称为 G13 问题,它拥有 3 个约束条件.初始种群包含 800 个个体,迭代进行 300 代.

$$\left. \begin{aligned} \min_x f(x) &= e^{x_1 x_2 x_3 x_4 x_5} \\ \text{s.t. } h_1(x) &= x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0 \\ h_2(x) &= x_2 x_3 - 5 x_4 x_5 = 0 \\ h_3(x) &= x_1^3 + x_2^3 + 1 = 0 \\ \text{The global minima: } x^* &= (-1.717143, 1.595709, 1.827247, -0.763641, -0.763645), \\ f(x^*) &= 0.0539498 \end{aligned} \right\} \quad (13)$$

当极坐标空间的半径固定为 $\sqrt{10}$ 时,约束条件 h_1 肯定可以满足.因此,当初始化和搜索在极坐标空间中进行时,所有个体的半径都会假定为 $\sqrt{10}$.在笛卡尔空间中,初始种群在超立方体 $x_i \in [-\sqrt{10}, \sqrt{10}], i=1,2, \dots, 5$ 中产生,初始种群的转换和 G3 问题相同.

G13 问题的罚函数设计见公式(14).类似于公式(12),公式(14)也具有偏离约束条件越严重的个体适应度越差的特点.

$$\text{fitness} = \begin{cases} f(x), & \text{if the individual is feasible} \\ \sum_i \text{abs}(h_i(x)) + e, & \text{otherwise} \end{cases} \quad (14)$$

4.2 实验结果分析

在实验中,G3 和 G13 问题分别运行了 20 次,得到的最优结果和平均结果见表 2.取 20 次之中最优结果的画

图,得到图 3~图 7.

从表 2 的结果中可以看出,10 维 G3 问题的 4 个设置的结果很接近.从图 3 的曲线中可以看出,S2 比 S1 收敛得快,S4 也比 S3 收敛迅速.这说明在极坐标空间得到的初始种群的质量比笛卡尔空间要好,使得启发式算法可以更快地收敛.此外,虽然 S3 和 S4 的最终结果最好,但是在图 3 中有陷入了局部极值的现象出现.这说明当目标问题的维数较低时,极坐标空间的优势不明显,但是在极坐标空间下初始化依然有不错的效果.

Table 2 Results of benchmark functions

表 2 性能评估函数优化结果

性能评估函数	S1	S2	S3	S4
G3, 10D (average)	0.999 994	0.999 998	1.000 000	1.000 000
G3, 10D (best)	0.999 996	0.999 999	1.000 000	1.000 000
G3, 40D (average)	0.980 600	0.983 687	0.985 237	0.986 748
G3, 40D (best)	0.995 381	0.995 885	0.995 653	0.998 345
G3, 70D (average)	0.611 867	0.684 801	0.835 390	0.886 304
G3, 70D (best)	0.821 512	0.840 857	0.864 586	0.945 874
G3, 100D (average)	0.234 862	0.409 831	0.782 416	0.825 901
G3, 100D (best)	0.313 861	0.425 624	0.855 211	0.879 551
G13 (average)	0.169 485	0.167 934	0.094 175 2	0.088 770 2
G13 (best)	0.054 173 7	0.054 009 1	0.054 309 5	0.053 958 6

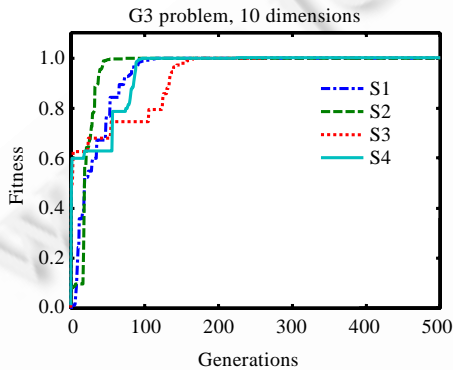


Fig.3 G3 problem, 10 dimensions

图 3 10 维 G3 问题

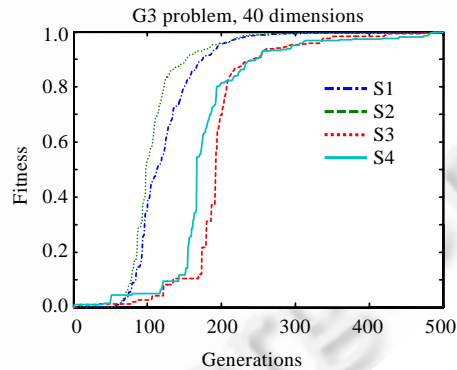


Fig.4 G3 problem, 40 dimensions

图 4 40 维 G3 问题

40 维 G3 问题的结果就与 10 维的结果有一些差别.虽然 S3 和 S4 在图 4 中依旧有陷入局部极值的现象,但是表 2 中平均结果和最优结果的差距已经比较明显,S3 和 S4 的结果要比 S1 和 S2 的好一些,结果之间的差距比 10 维的大.

在 70 维 G3 问题中,S4 的结果明显比其他 3 个设置要好,收敛速度也更快.从图 5 看,S2 和 S3 差不多,劣于 S4,但是比原始算法 S1 要好不少.此外,S3 搜索得到的平均结果比 S2 好很多.这说明基于极坐标映射的启发式算法很适合处理这类约束条件,虽然在低维空间中效果不明显,但是在高维空间中的算法性能大幅度提高.

当 G3 问题增加到 100 维后,从图 6 中很明显可以看出:在笛卡尔空间中,迭代搜索的 S1 和 S2 难以找到较好的解;而在极坐标空间下,迭代搜索的 S3 和 S4 都能保持着较好的性能.这说明基于极坐标映射的启发式算法受空间维数的影响较小.

图 7 是 G13 问题的结果.S4 找到的结果最好,并且 S3 和 S4 都使用较少的迭代次数找到最优解,比 S1 和 S2 要快很多.从最终结果的平均值来看,S3 和 S4 的性能更稳定一些.不过,极坐标空间初始化在性能上的差别并不明显.这是因为 G13 问题的维数较少,基于极坐标映射的启发式算法的优势并不如在高维度时明显,初始种群的质量相差不大.

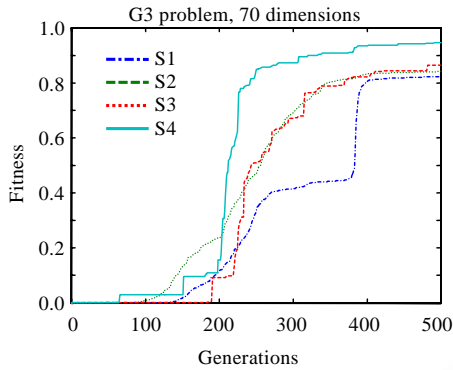


Fig.5 G3 problem, 70 dimensions
图 5 70 维 G3 问题

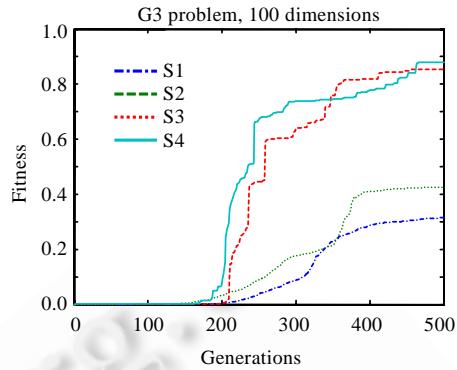


Fig.6 G3 problem, 100 dimensions
图 6 100 维 G3 问题

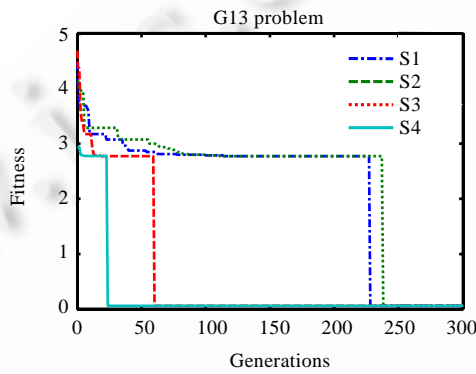


Fig.7 G13 problem
图 7 G13 问题

根据上述性能评估实验的结果,本文提出的基于极坐标映射的启发式算法在具有特定约束条件的最优化问题中有很好的效果.在所有的实验中,基于极坐标映射的启发式算法搜索到的结果比HGAPSO算法好,并且目标问题的维数越高,性能的差距越大.虽然在低维空间中,基于极坐标映射的启发式算法陷入局部极值的概率较大,但是最终都能跳出局部极值,优化得到的结果很优秀.实验结果和理论分析吻合.

5 P2P 网络服务器部署方案实验

基于极坐标映射的启发式算法最后被用于 P2P 网络服务器的优化部署问题上.在该问题中,区域数量 n 是由网络拓扑决定的,一般不会改变,全局服务器的数量 G 也可以认为是一个固定值,因此,P2P 网络服务器部署问题可视为在 n 维空间中寻找一个满足约束条件的目标函数值最优的点.具体的优化目标函数见公式(5).其约束条件可以使用类似于 G3 问题的解决办法,即使用基于极坐标映射的启发式算法.

假设需要优化的 P2P 网络含有 $n=11$ 个大小不等的区域,区域的节点数量构成一个从 50 万~100 万的等差数列.根据在 2000 年关于 Gnutella 网络搭便车现象的观察研究^[16],P2P 网络的搭便车现象十分严重,高达 70% 的节点都是搭便车节点.后续的研究也表明,这种现象没有改善.所以,本实验假设所有区域的初始搭便车节点比例均为 50%.每个区域的区域级服务器数量为 400,全局级服务器数量 $G=4000,k=50$ 个时间段.与贡献和获取有关的常数 $\alpha=2,\beta=1,\gamma=1000$,即在每个时间段中,平均每个节点可以上传 1 单位资源,并试图获取 2 单位资源,服务器的上传能力为普通节点的 1000 倍.最大代价常数 $\delta_{\max}=2$,即当节点的付出是回报的 2 倍时,所有节点都会转变为搭便车节点.比例常数 $c=100$ 万,即每个区域的目标规模在 100 万左右.

初始种群中包含 400 个个体,迭代进行 500 代,S1~S4 的设定和性能评估实验相同.笛卡尔空间中使用的罚函数类似于 G3 问题.当个体满足约束条件时,适应度为目标函数值;当个体不满足约束条件时,个体适应度为负数,并且个体距离可行解域越远,适应度越低.此外,为了更好地体现优化部署的效果,实验设计一个性能对比方案 S5:根据各个区域现有节点数将全局级服务器按比例进行分配.

对 P2P 网络服务器部署问题进行了 20 次实验,得到的结果见表 3.取 20 次之中最好的结果画图,得到图 8.由于方案 S5 不存在随机性,也没有迭代过程,只列出计算结果.

Table 3 Results of P2P network server allocation problem

表 3 P2P 网络服务器部署问题结果

	S1	S2	S3	S4	S5
Average solution	2.110×10^6	2.191×10^6	2.311×10^6	2.311×10^6	None
Best solution	2.224×10^6	2.251×10^6	2.322×10^6	2.322×10^6	2.086×10^6

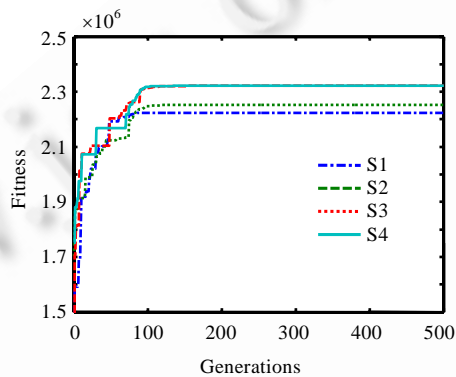


Fig.8 P2P network server allocation problem

图 8 P2P 网络服务器部署问题

由于 P2P 服务器部署问题的维数不高,类似于 10 维和 40 维的 G3 问题,虽然 S3 和 S4 在优化的过程中有陷入局部最优的情况出现,但是最终的结果要优于 S1 和 S2,并且在极坐标空间下初始化的 S2 得到的结果比对应的 S1 要好一些.

从表 3 可以看出,通过启发式算法优化得到的部署方案对比方案 S5 要好,同样数量的服务器经过合理部署后,可以使得网络中包含更多的正常节点.P2P 网络服务器管理者可以利用该方案部署服务器,将一部分服务器部署在与区域无关的位置中,每隔一段时间改变这些服务器的设置,让他们为不同的区域提供服务,满足更多节点的需求,提升 P2P 网络的吸引力.

6 结 论

为了更好地利用服务器满足 P2P 网络中节点的需求、提高 P2P 网络的性能以吸引更多节点加入,本文提出了一种 P2P 网络服务器最优部署方案,将服务器分为两级进行部署,通过只调整全局级服务器的设置,使其优先为某些区域提供服务,使得网络的整体性能提高.该部署方案是一个带约束条件的最优化问题,本文为此提出了一种基于极坐标映射的启发式算法来解决这个问题.由于与长度相关的约束条件很适合在极坐标空间中表达,因此极坐标映射可以为启发式算法带来很多好处,例如减小搜索空间、去除一些搜索边界、提高初始种群的质量等等.有了这些优势,启发式算法可以用更短的时间找到更好的结果.

在性能评估实验中,以 HGAPSO 算法作为对比,对两个常见的性能评估函数进行了验证.实验结果表明,基于极坐标映射的启发式算法在性能和结果上有优势,并且这个优势随着目标函数的维数增加而增大.最后,该算法被用于对 P2P 网络服务器部署方案的优化,结果表明,合理部署服务器资源可以使得 P2P 网络吸引更多节点

加入。

References:

- [1] Hei XJ, Liang C, Liang J, Liu Y, Ross KW. A measurement study of a large-scale P2P IPTV system. *IEEE Trans. on Multimedia*, 2007,9(8):1672–1687. [doi: 10.1109/TMM.2007.907451]
- [2] Jiang H, Li J, Li ZC, Bai XY. Performance evaluation of content distribution in hybrid CDN-P2P network. In: *Proc. of the 2nd Int'l Conf. on Future Generation Communication and Networking (FGCN 2008)*. 2008. 188–193. [doi: 10.1109/FGCN.2008.63]
- [3] Wu A, Liu XS, Fu QY, Liu KJ. DPVoD: P2P based video-on-demand architecture. *Journal of Computer Research and Development*, 2008, 45(2):269–277 (in Chinese with English abstract).
- [4] Spoto S, Gaeta R, Grangetto M, Sereno M. Analysis of PPLive through active and passive measurements. In: *Proc. of the IEEE Int'l Symp. on Parallel & Distributed Processing (IPDPS 2009)*. 2009. 1–7. [doi: 10.1109/IPDPS.2009.5160956]
- [5] Damousis IG, Bakirtzis AG, Dokopoulos PS. Network-Constrained economic dispatch using real-coded genetic algorithm. *IEEE Trans. on Power Systems*, 2003,18(1):198–205. [doi: 10.1109/TPWRS.2002.807115]
- [6] Eberhart R, Kennedy J. A new optimizer using particle swarm theory. In: *Proc. of the 6th Int'l Symp. on Micro Machine and Human Science (MHS'95)*. 1995. 39–43. [doi: 10.1109/MHS.1995.494215]
- [7] Kennedy J, Russell E. Particle swarm optimization. In: *Proc. of the IEEE Int'l Conf. on Neural Networks, Vol.4*. 1995. 1942–1948. [doi: 10.1109/ICNN.1995.488968]
- [8] Kao YT, Zahara E. A hybrid genetic algorithm and particle swarm optimization for multimodal functions. *Applied Soft Computing*, 2008,8(2):849–857. [doi: 10.1016/j.asoc.2007.07.002]
- [9] Settles M, Soule T. Breeding swarms: A new approach to recurrent neural network training. In: *Proc. of the 2005 Conf. on Genetic and Evolutionary Computation*. 2005. 185–192. [doi: 10.1145/1068009.1068038]
- [10] Juang CF. A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2004,34(2):997–1006. [doi: 10.1109/TSMCB.2003.818557]
- [11] Matthysen W, Engelbrecht AP. A polar coordinate particle swarm optimiser. *Applied Soft Computing*, 2011,11(1):1322–1339. [doi: 10.1016/j.asoc.2010.04.005]
- [12] Sierra A, Echeverria A. The polar evolution strategy. In: *Proc. of the IEEE Congress on Evolutionary Computation (CEC 2006)*. 2006. 2301–2306. [doi: 10.1109/CEC.2006.1688592]
- [13] Feldman M, Papadimitriou C, Chuang J, Stoica I. Free-Riding and whitewashing in peer-to-peer systems. *IEEE Journal on Selected Areas in Communications*, 2006,24(5):1010–1019. [doi: 10.1109/JSAC.2006.872882]
- [14] Shi YH, Eberhart R. A modified particle swarm optimizer. In: *Proc. of the IEEE Int'l Conf. on Evolutionary Computation, World Congress on Computational Intelligence*. 1998. 69–73.
- [15] Test functions of constrained global optimization. http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm
- [16] Eytan A, Huberman BA. Free-Riding on gnutella. <http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/viewArticle/792.HUMPHREY>

附中文参考文献:

- [3] 吴艾,刘心松,符青云,刘克剑.DPVoD:基于 P2P 的视频点播体系结构. *计算机研究与发展*,2008,45(2):269–277.



曾明霏(1984—),男,广西南宁人,博士生,主要研究领域为覆盖网与用户行为研究.
E-mail: zz_fei@hotmail.com



余顺争(1958—),男,博士,教授,博士生导师,CCF 会员,主要研究领域为网络安全,网络行为分析.
E-mail: syu@mail.sysu.edu.cn